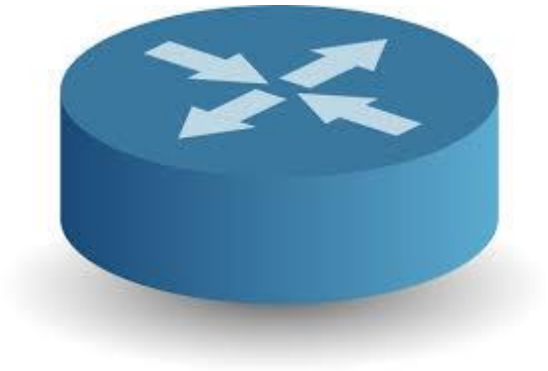


CISCO ESZKÖZ BEÁLLÍTÁSA 2018.11.05.

Kivonat

A Cisco router és switch parancsainak összefoglalása

Virágh Krisztián

Tartalomjegyzék

Router beállítása:
felhasználói mód: gyakorlas>
privilegizált mód: gyakorlas#
globális konfigurációs mód: gyakorlas(config)#

Hostname beállítás

A router neve legyen a gyakorlas!

```
Router>enable
Router#configure terminal
Router(config)#hostname gyakorlas
gyakorlas(config)#
```

Jelszavak beállítása

Konzol jelszó

A konzol jelszó beállítása 123-ra.

```
gyakorlas(config)#line console 0
gyakorlas(config-line)#password 123
gyakorlas(config-line)#login
gyakorlas(config-line)#
```

Privilegizált jelszó

Privilegizált jelszó beállítása 456-ra

```
gyakorlas(config)#enable password 456
gyakorlas(config)#
```

Ha mindezt titkosítvca szeretném

```
gyakorlas(config)#enable secret 456
gyakorlas(config)#
```

VTY jelszavak

VTY jelszavak beállítása 789-re

```
gyakorlas(config)#line vty 0 4
gyakorlas(config-line)#password 789
gyakorlas(config-line)# exit
gyakorlas(config)# exit
gyakorlas#
```

Jelszavak titkosítása

```
gyakorlas(config)# service password-encryption
```

Banner üzenet

Bejelentkezési üzenet

```
gyakorlas(config)#banner login #üzenet#
```

Napi üzenet

```
gyakorlas(config)#banner motd #üzenet#
```

A Switch-en csak ezt lehet létrehozni

Ip címek beállítása

Ethernet interfész IPV4 címének beállítása

fastethernet ip címének beállítása 192.168.0.14/24-re

```
gyakorlas#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/0 unassigned YES unset administratively down down
FastEthernet0/1 unassigned YES unset administratively down down
Vlan1 unassigned YES unset administratively down down
gyakorlas#
gyakorlas#configure terminal
gyakorlas(config)#interface fastethernet0/0
gyakorlas(config-if)#ip  address  192.168.0.14
255.255.255.0
gyakorlas(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
gyakorlas(config-if)#
gyakorlas (config-if)#duplex auto | half | full
gyakorlas (config-if)#speed auto | 10 | 100
Gyakorlas(config-if)#exit
```

(Switch) VLAN1 IPV4 címének beállítása

SWITCH ESETÉN: VLAN1 ip címének beállítása 192.168.0.214-re

```
Switch#show ip interface brief
Interface IP-Address OK? Method Status Protocol
FastEthernet0/1 unassigned YES manual down down
                .
                .
                .
Vlan1 unassigned YES manual administratively down down
Switch#
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface vlan1
Switch(config-if)#ip address 192.168.0.214 255.255.255.0
Switch(config-if)#
Switch(config-if)#no shutdown
%LINK-5-CHANGED: Interface Vlan1, changed state to up
Switch#show ip interface brief
```

Interface IP-Address OK? Method Status Protocol
FastEthernet0/1 unassigned YES manual down down
.
.
.
.
FastEthernet0/24 unassigned YES manual down down
Vlan1 192.168.0.214 YES manual up down
Gyakorlas(config-if)#exit

Soros interfész IPV4 címének beállítása

```
Gyakorlas(config)#interface serial 0/1/0
Gyakorlas(config-if)#ip address 172.16.17.19
255.255.255.127
Gyakorlas(config-if)#description Keszrouter_fele
Gyakorlas(config-if)#clock rate 64000
Gyakorlas(config-if)#no shutdown
Gyakorlas(config-if)#exit
```

Ethernet interfész IPV6 címének beállítása

```
Router(config)#interface GigabitEthernet0/0
Router(config-if)#ipv6 address fc001::1/64
Router(config-if)# clock rate 128000
Router(config-if)#no shutdown
```

ellenőrzés:

```
Router#show ipv6 interface brief
```

GigabitEthernet0/0 [up/up]
FE80::260:47FF:FEA0:9201
FC00::1
GigabitEthernet0/1 [administratively down/down]

Loopback interface beállítás

```
Router(config)#interface loopback 0
Router(config-if)# ip address 200.0.0.1 255.255.255.255
```

Konfiguráció ellenőrzése

Futó konfiguráció

```
gyakorlas#show running-config
```

Induló konfiguráció

```
gyakorlas#show startup-config
```

Beállítások mentése

Beállítások mentése a készüléken

```
gyakorlas#write
```

Building configuration...

[OK]

Vagy:

gyakorlas#copy runing-config startup-config

Building configuration...

[OK]

Beállítások törlése

gyakorlas#erase startup-config

R1#erase startup-config
R1#delete nvram:startup-config
R1#del nvram:startup-config
R1#write erase
R1#wr erase
R1#reload

Ha van VLan:

delete flash:vlan.dat

Beállítások biztonsági másolata TFTP szerveren

tftp : trivial ftp udp-n keresztül, nem garantálja az átvitel sikerét.
tftp-udp
csak közeli (helyi) hálózaton
nem kezel könyvtárakat
nem kezel jogosultságokat
max 32 Mb

MENTÉS

```
PEZOdemo(config)#do write
Building configuration...
PEZOdemo#copy flash: tftp:
Source filename []? c1900-universalk9-mz.SPA.151-4.M4.bin
Address or name of remote host []? 192.168.100.200
Destination filename [c1900-universalk9-mz.SPA.151-
4.M4.bin]? fontos.bin

Writing c1900-universalk9-mz.SPA.151-
4.M4.bin...!!!!!!!!!!!!!!!!!!!!!!!!
[OK - 33591768 bytes]

33591768 bytes copied in 0.839 secs (4203810 bytes/sec)
```

TÖRLÉS

```
PEZOdemo#delete flash:c1900-universalk9-mz.SPA.151-4.M4.bin
Delete filename [c1900-universalk9-mz.SPA.151-4.M4.bin]?y
```

Delete flash:/y? [confirm]y%Error deleting flash:/y (No such file or directory)
PEZOdemo#reload

VISSZAÁLLÍTÁS TFTP-RŐL

Router>enab
Router#conf t
System flash directory:
File Length   Name/status
  3   33591768 c1900-universalk9-mz.SPA.151-4.M4.bin
  2   28282   sigdef-category.xml
  1   227537   sigdef-default.xml
[33848212 bytes used, 221895788 available, 255744000 total]
249856K bytes of processor board System flash (Read/Write)
Router(config)#do delete flash:c1900-universalk9-mz.SPA.151-4.M4.bin
Delete filename [c1900-universalk9-mz.SPA.151-4.M4.bin]?
Delete flash:/c1900-universalk9-mz.SPA.151-4.M4.bin? [confirm]

Router(config)#do reload
Proceed with reload? [confirm]
System Bootstrap, Version 15.1(4)M4, RELEASE SOFTWARE (fc1)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 2010 by cisco Systems, Inc.
Total memory size = 512 MB - On-board = 512 MB, DIMM0 = 0 MB
CISCO1941/K9 platform with 524288 Kbytes of main memory
Main memory is configured to 64/-1(On-board/DIMM0) bit mode with ECC disabled

Readonly ROMMON initialized

Boot process failed...

The system is unable to boot automatically. The BOOT environment variable needs to be set to a bootable image.
rommon 1 > tftpdnld

Missing or illegal ip address for variable IP_ADDRESS
Illegal IP address.

usage: tftpdnld
  Use this command for disaster recovery only to recover an image via TFTP.
  Monitor variables are used to set up parameters for the transfer.

```
  (Syntax: "VARIABLE_NAME=value" and use "set" to show
current variables.)
  "ctrl-c" or "break" stops the transfer before flash erase
begins.

  The following variables are REQUIRED to be set for
tftpdnld:
            IP_ADDRESS: The IP address for this unit
         IP_SUBNET_MASK: The subnet mask for this unit
        DEFAULT_GATEWAY: The default gateway for this unit
           TFTP_SERVER: The IP address of the server to
fetch from
             TFTP_FILE: The filename to fetch

  The following variables are OPTIONAL:
          TFTP_VERBOSE: Print setting. 0=quiet,
1=progress(default), 2=verbose
      TFTP_RETRY_COUNT: Retry count for ARP and TFTP
(default=7)
           TFTP_TIMEOUT: Overall timeout of operation in
seconds (default=7200)
          TFTP_CHECKSUM: Perform checksum test on image,
0=no, 1=yes (default=1)
          FE_SPEED_MODE: 0=10/hdx, 1=10/fdx, 2=100/hdx,
3=100/fdx, 4=Auto(deflt)

rommon 5 > ?
boot                boot up an external process
confreg             configuration register utility
dir                 list files in file system
help                monitor builtin command help
reset               system reset
set                 display the monitor variables
tftpdnld            tftp image download
unset               unset a monitor variable
rommon 6 > IP_ADDRESS=192.168.100.100
rommon 7 > IP_SUBNET_MASK=255.255.255.0
rommon 8 > DEFAULT_GATEWAY=10.10.10.10
rommon 9 > TFTP_SERVER=192.168.100.200
rommon 12 > TFTP_FILE=fontos.bin
rommon 13 > tftpdnld

           IP_ADDRESS: 192.168.100.100
        IP_SUBNET_MASK: 255.255.255.0
       DEFAULT_GATEWAY: 10.10.10.10
           TFTP_SERVER: 192.168.100.200
             TFTP_FILE: fontos.bin
Invoke this command for disaster recovery only.
WARNING: all existing data in all partitions on flash will
be lost!
```

```
Do you wish to continue? y/n:  [n]:  y

.....[TIMED OUT]
TFTP: Operation terminated.
```

rommon 14 > A TIE OUT kijavítása: A swicth portját fastport (végponti) beállításra kell konfigurálni.

```
Switch(config-if)#spanning-tree portfast
```

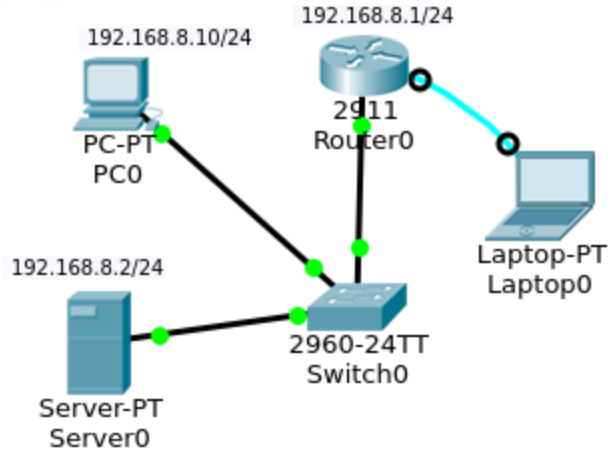
Beállítások biztonsági másolata FTP szerveren

```
DEMO#conf t
DEMO(config)#ip ftp username cisco
DEMO(config)#ip ftp password class
DEMO(config)#exit
DEMO#
%SYS-5-CONFIG_I: Configured from console by console

DEMO#copy flash: ftp:
Source filename []? c1900-universalk9-mz.SPA.151-4.M4.bin
Address or name of remote host []? 192.168.100.200
Destination filename [c1900-universalk9-mz.SPA.151-4.M4.bin]?
demo_ios.bin

Writing c1900-universalk9-mz.SPA.151-4.M4.bin...
[OK - 33591768 bytes]
```

NTP szerver beállítása

```
server
R1#clock set 20:05:52 10 sept 2020
R1(config)#clock timezone HU 2 0
R1(config)#ntp master
R1(config)#ntp authentication-key 1 md5 NTPpa55
R1(config)#ntp trusted-key 1
R1(config)#ntp authenticate

kliens
R2(config)# ntp server 192.168.100.1
R2(config)# ntp authenticate
R2(config)# ntp trusted-key 1
R2(config)# ntp authentication-key 1 md5 NTPpa55
R2(config)# ntp update-calendar
R2(config)#service timestamps log datetime msec

ellenőrzés
R1#show clock details
R1#show clock
R1#show ntp status
R1#show ntp associations
```

Naplózás



Különféle naplózási lehetőségek

Öt különféle naplózás:

- Console logging
- Terminal logging (VTY)
- Buffered logging (RAM)
- Syslog Server logging (távoli szerver)
- SNMP trap logging (SNMP szerver)

A célunk egy SSH távoli hozzáférés beállítása, hogy legyen mit naplózzunk. Konkrétan a sikertelen próbálkozásokat szeretnénk látni.

SSH beállítás:

```
Router(config)#h R1
R1(config)#ip domain-name span.com
R1(config)#crypto key generate rsa

R1(config)#username joska secret titok
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#transport input ssh
R1(config-line)#exit
```

A naplózáshoz állítsuk be a rendszeridőt.

```
R1#clock set 20:30:00 25 Sep 2017
```

Az idő beállítása nélkül a naplóállományokban hamis adatok jelennek meg.

Naplózás konzolra

Kapcsoljuk be a naplózást:

```
R1#conf t
R1(config)#logging on
```

Naplózzuk a sikertelen SSH belépési kísérleteket:

```
R1(config)#login on-failure log
```

A konzolra naplózás be van kapcsolva alapértelmezetten. Ha még sem, akkor kapcsoljuk be:

```
R1(config)#logging console
```

Ellenőrizzük. Próbáljunk meg belépni SSH protokollon keresztül, rossz jelszót megadva.

Cisco Packet Tracer PC parancssorából:

```
ssh -L janos 192.168.10.1
```

Az eredmény ehhez hasonló lehet:

```
R1(config)#%SEC_LOGIN-5-LOGIN_FAILED: Login failed
[user: janos] [Source: 192.168.10.51] [localport: 22]
[Reason: Login Authentication Failed] at 00:25:38
UTC H márc. 1 2017
```

Konzolra naplózás kikapcsolása:

```
R1(config)#no logging console
```

Naplózás távoli syslog szerverre

Naplózás engedélyezése:
```
R1(config)#logging on
```

Naplózzuk a sikertelen SSH belépési kísérleteket:
```
R1(config)#login on-failure log
```

Távoli syslog szerver megadása:
```
R1(config)#logging host 192.168.8.1
```

A naplószerveren a következő üzenetet látjuk:
```
%SEC_LOGIN-5-LOGIN_FAILED:
Login failed [user: janos]
```

```
[Source: 192.168.10.51] [localport: 22]
[Reason: Login Authentication Failed]
at 20:25:24 UTC H szept. 25 2017
```

Egyéb lehetőségek

Időbélyeg beállítása naplózáshoz:

```
R1(config)# service timestamps log datetime msec
R1(config)# service timestamps debug datetime msec
```

A naplófájlok mellé beállítja az időpontot.

Naplózási beállítások:

```
R1(config)# no logging console
R1(config)# no logging buffered
```

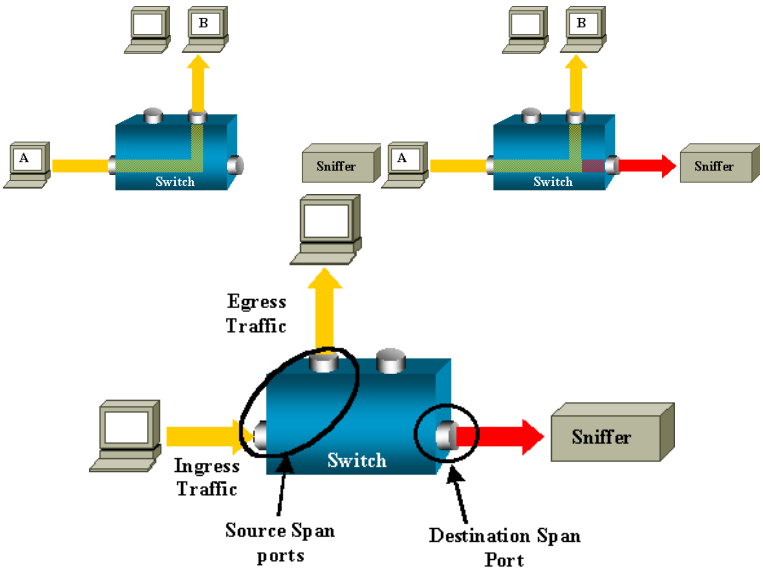
Naplóbeállítások megtekintése:

```
R1# show logging
```

Belépések naplózása:

```
R1(config)#login on-failure log
R1(config)#login on-success log
R1(config)#logging trap debug //debugéok nssplózáds
```

SPAN port készítése




```
Switch(config)# monitor session 1 source interface
gigabitethernet0/1
Switch(config)# monitor session 1 destination interface
gigabitethernet0/2 encapsulation replicate
Switch(config)# end
```

```
Switch(config)# no monitor session 1
```

Távelérés SSH-val

SSH elérés beállítása Routeren

```
Router(config)#hostname router1
Router(config)#ip domain-name teszt.hu
Router(config)#crypto key generate rsa  //Ajánlott az 1024 bit
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
```

vagy:

```
Router(config)#crypto key generate rsa general-keys modulus 1024
Router(config)#ip ssh version 1 | 2
Router(config)#ip ssh time-out 60        //mp-ben megadva
Router(config)#ip ssh authentication-retries 2
Router(config)#username felhasználó neve password jelszava
Router(config)#username felhasználó neve password jelszava
Router(config)#username felhasználó neve password jelszava
//Több felhasználót is létrehozhatok
Router(config)#security passwords min-length 12
//minimális jelszó jossz
Router(config)#line vty 0 15
Router(config-line)#login local
Router(config-line)#transport input ssh
Router(config-line)#line vty 0 15 exec timeout 2 30
 //2 perc 30 másodperc elteltével megszakad a kapcsolat
Router(config)# login block-for 100 attempts 2 within 50
//100 másodpercig megakadályozza a belépést, ha 50 mp alatt 2-szer elrontottam a felhasználó név – jelszó párost
Router(config)# login delay 10
//az egymást követő bejelentkezési kísérletek között eltelt idő mp-ben
Router (config)#ip ssh time-out 10
//mennyi időn belül kell beírnom a jó jelszót (mp)
Router (config)#ip ssh authentication-retries 1
//hányszór írhatom be a jelszót
Router(config-line)#privilege level 15
```

Kulcs törlése:

```
Router(config)#crypto key zeroize rsa
```

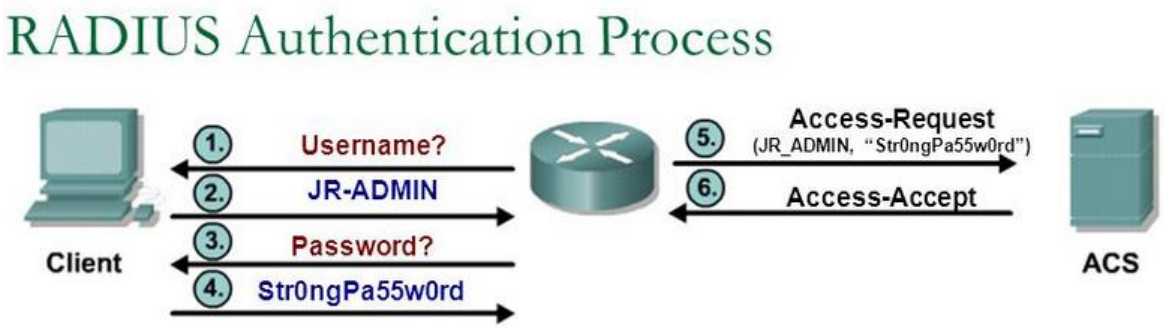
Belépés SSH-val

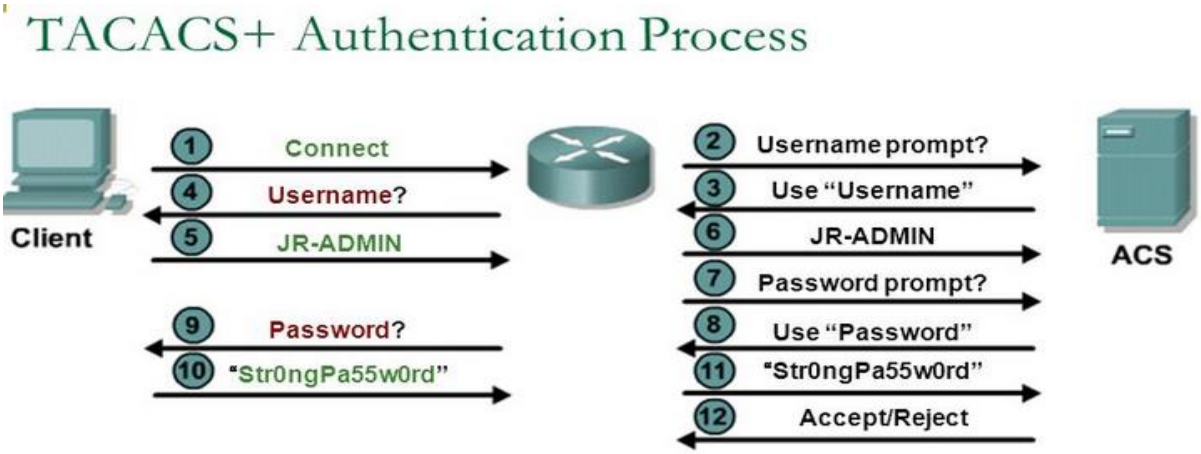
```
ssh -l felhasznalo_nev 192.168.100.1
password:
```

…

```
exit
```

Hitelesítés RADIUS vagy TACACS serverrel





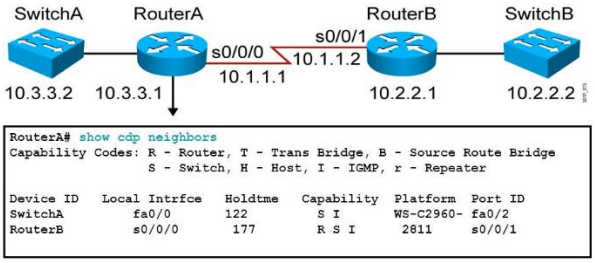

```
Switch(config)# ip routing
Switch(config)# aaa new-model
Switch(config)# username admin secret MyPassword
Switch(config)# radius-server | tacacs-server host 10.0.0.100
auth-port 1812 acct-port 1813 key MyRadiusKey

Switch(config)# aaa authentication dot1x default group radius

Switch# show dot1x interface g0/12
Supplicant MAC
   AuthSM State       = N/A
   BendSM State       = N/A
PortStatus          = N/A
MaxReq              = 2
MaxAuthReq          = 2
HostMode            = Single
PortControl         = Auto
QuietPeriod         = 60 Seconds
Re-authentication   = Disabled
ReAuthPeriod        = 3600 Seconds
ServerTimeout       = 30 Seconds
SuppTimeout         = 30 Seconds
TxPeriod            = 30 Seconds
```

Ha a belépéshez kell a hitelesítés:

```
Switch(config)#line vty 0 4
Switch(config-line)#login authentication default
```

Hálózat felderítése

Router#show cdp neighbors
//a közvetlenül az eszközömhöz csatlakozott eszközök listázása



Router#show cdp neighbors details
//részletes lista

Elfelejtett jelszó helyreállítása

ROUTER

A jelszóval védett rendszereken a jelszó ismerete nélkül nem lehet hozzáférni a router privilegizált promptjához, nem lehet érdemi munkát végezni rajta. Elfelejtett jelszó esetén, amennyiben van

1. közvetlen konzol kapcsolatunk, és
2. hozzáférünk a forgalomirányító tápellátásához

a privilegizált prompt mégis elérhető. Az elréshez biztosítanunk kell, hogy a router ne töltse be a konfigurációt. Ehhez a konfigurációs regiszterben ki kell kapcsolni a betöltést vezérlő bitet.

1. Indítsuk újra a routert, a tápellátás ki- illetve bekapcsolásával. Ügyeljünk arra, hogy a hirtelen áramlökéseket elkerüljük. Az elektromos eszközöket nem szabad azonnal visszakapcsolni - várjunk néhány másodpercet (5 - 10 sec).
2. A betöltési folyamatot miután a forgalomirányító jelzi az IOS betöltését (#####...) szakítsuk meg **Ctrl + Break** vagy **Ctrl + C** billentyűk lenyomásával. Ekkor ROM monitor módba kerülünk.
   Itt kicsit másként viselkedik a parancssoros környezet. A segítség kéréséhez a **?** után Entert kell ütnünk.
   rommon 1 >?
   alias set and display aliases command
   boot boot up an external process
   break set/show/clear the breakpoint
   confreg configuration register utility
   context display the context of a loaded image
   dev list the device table
   dir list files in file syste
3. A **conreg** parancsot kell kiadni, ami után hexadecimális módon be kell írni a konfigurációs regiszter új értékét. Ezt úgy állítjuk be, hogy konfigurációs file nélkül induljon el a rendszer.

rommon 2 > confreg 0x2142
A 4-es érték felel a konfigurációs file kikapcsolásáért!

4. Indítsuk újra a routert monitor módban a
rommon 3 >reset
parancs kiadásával! Az IOS betöltődik és megkérdezi, hogy akarjuk-e a konfigurációs dialógust választani a rendszer beállításához? Válaszoljunk no-val, majd erősítsük meg yes-el, hogy ténylegesen nem akarjuk ennek elindulását!
5. Lépjünk be privilegizált módba és nézzük meg a **startup-config** beállításait! Eldönthetjük, hogy szükséges-e jelszót cserélni. Ez szükséges, ha
    1. van secret jelszó, illetve
    2. be van állítva a jelszó titkosítás minden jelszóra (service password-encryption).

    Amennyiben szükséges töltsük át a konfigurációt a memóriába
    copy startup-config running-config (sh flash -> startup ios neve)
    lépjünk be globális konfigurációs módba és állítsuk be a jelszavakat a szokásos módon!
6. Engedélyezzük az interfészeket!
7. Változtassuk vissza a konfigurációs regiszter értékét, hogy a következő induláskor már ismét betöltse a konfigurációt!
    config-register 0x2102
    Ellenőrizzük ennek sikerességét! Ekkor a **show version** parancs utolsó sorában a következőt kell látnunk
    Configuration register is **0x2142** (will be **0x2102** at next reload).
8. Mentsük a beállításokat
    copy running-config startup-config
    és indítsuk újra a forgalomirányítót
    reload

SWITCH

1. Terminál emulátor (pl.Hyper Terminal) soros konzol portjának beállítása az alábbi értékekre:
Sebesség: 9600 bit/sec
Adat bit: 8
Paritás: Nincs
Stop bitek száma: 1
Folyamatvezérlés: Xon/Xoff

2. Terminál soros konzol portjának csatlakoztatása a Cisco switch eszközhöz.
3. Cisco tápcsatlakozójának kihúzása
4. Mode gomb lenyomása mellett a tápcsatlakozó visszadugása a switchbe. Miután az 1x port feletti LED már nem világít rá 1-2 másodperc után a mode gomb elengedhető.

5. Ezután a következő utasítások jelennek meg:

```
...
The system has been interrupted prior to initializing the
flash file system.
The following commands will initialize the flash file system,
and finish
loading the operating system software:
flash_init
load_helper
boot
```

6. Gépelje **be:flash_init**
7. Gépelje be:**load_helper**
8. Gépelje be:**dir flash:**

A switch file rendszer megjelenik:

```
Directory of flash:
2 -rwx 843947 Mar 01 1993 00:02:18 C2900XL-h-mz-112.8-SA
4 drwx   3776 Mar 01 1993 01:23:24 html
66 -rwx    130 Jan 01 1970 00:01:19 env_vars
68 -rwx 1296   Mar 01 1993 06:55:51 config.text
1728000 bytes total (456704 bytes free)
```

9. Gépelje be: **rename flash:config.text flash:config.old**, amivel a config.text konfigurációs fájlt átnevezi config.old névre.

```
  This file contains the password definition.
```

10. Gépelje be: **boot** a rendszer újraindításához.
11. Gépelje be: **N** a promptnál hogy a a setup program elindulásakor.

```
Continue with the configuration dialog? [yes/no] : N
```

12. switch prompt megjelenésekor Gépelje be: **en** az enabled módba lépéshez.
13. Gépelje be: **flash:config.old flash:config.text** amivel visszanevezi a konfigurációs fájlt az eredeti nevére.
14. A konfigurációs fájl memóriába töltése:

```
Switch#copy flash:config.text system:running-config
Source filename [config.text]? <cr>
Destination filename [running-config]? <cr>
```

A konfigurációs fájl memóriába töltődik:
15. Jelszó kicserélése:

```
switch#configure terminal
switch(config)#no enable secret
```

!--> Ha a switch enable secret jelszóval rendelkezik:

```
switch(config)#enable password Cisco
```

16. A futásidejű konfiguráció visszaírása a konfigurációs fájlba:

```
switch#write memory
```

Port-security

Portbiztonság konfigurálása

```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
//Ezzel bekapcsolom a port-biztonságot
Switch(config-if)#switchport port-security mac-address sticky
//„Ragadós” MAC address megjegyzi az első címet
Switch(config-if)#switchport port-security mac-address
0123.4567.89AB //vagy általunk megadott címmel:
Switch(config-if)#switchport port-security maximum 1
//Hány címet jegyezen meg
Switch(config-if)#switchport port-security violation shutdown
//Lekapcsolja a portot megsértés esetén
Switch(config-if)#switchport port-security violation [ protect
restrict ]
//ha nem szeretnénk, hogy letiltson:
protect – védelem, restricz - korlátozás
```

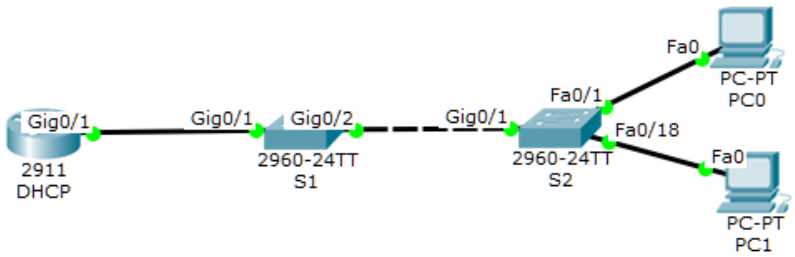
Portbiztonság miatt letiltott port újraengedélyezése

```
Switch(config)#int fa0/1
Switch(config-if)#shutdown
Switch(config-if)#no shut
```

DHCP Snooping

Az illetéktelen DHCP válaszok megakadályozása a nem megbízható portokon.

Példa:




```
S1(config)#ip dhcp snooping | vlan 10,20, 15-18
//DHCP Snooping globális engedélyezése
S1(config)#no ip dhcp snooping information option
//Ez kell ahhoz, hogy a snooping több switchen keresztül is
működjön
S1(config)#interf gi0/1
S1(config-if)#ip dhcp snooping trust
//Megbízható port kijelölése (a porton kell megadni)
S2(config)#ip dhcp snoop
S2(config)#no ip dhcp snoo infor opti
```

```
S2(config)#interf gig0/1
S2(config-if)#ip dhcp snoop trust
```

```
Switch(config-if)#ip dhcp snooping limit rate 10
//Nem megbízható porton a DHCP kérések limitje
```

VLAN-ok használata

VLAN-ok létrehozása

Első módszer:

```
Switch#vlan database
Switch(vlan)#vlan 10 name alfa
VLAN 10 added:
Name: alfa
Switch(vlan)#vlan 100 name beta
VLAN 100 added:
Name: beta
```

Második módszer:

```
Switch(config)#vlan 25
Switch(config-vlan)#name gamma
```

Portok hozzárendelése adott VLAN-hoz

```
Switch(config)#int fa0/1
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
```

Egyszerre több port hozzárendelése

```
Switch(config)#int range fa0/10 - 15
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 25
```

Trönkport beállítása

```
Switch(config)#int fa0/24
Switch(config-if)#switchport mode trunk
```

Natív VLAN beállítása

a trönk mindkét végén meg kell adni!

```
Switch(config-if)#switchport trunk native vlan 99
```

Engedélyezett VLAN-ok megadása a trönkön

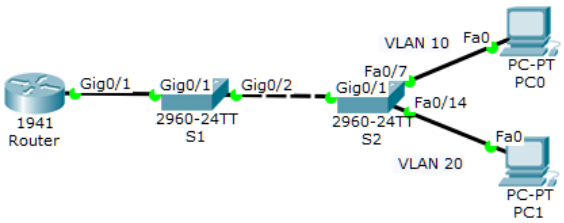
```
Switch(config-if)#switchport trunk allowed vlan [vlan_száma 2|
vlanok_száma felsorolva pl:3,4 |  vagy all]
```

```
//Ezt a trönk port interfészen kell megadni.
```

Trönk állapotának ellenőrzése

```
Switch# show interfaces trunk
```

DHCP snooping valn-okon




```
Router(config)#interf gig0/1.1
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interf gig0/1.2
Router(config-subif)#encapsulation dot1Q 20
Router(config-subif)#ip address 192.168.20.1 255.255.255.0
Router(config-subif)#exit
Router(config)#interf gig0/1
Router(config-if)#no ip add
Router(config-if)#no shu
Router(config-if)#exit
Router(config)#ip dhcp pool 10
Router(dhcp-config)#network 192.168.10.0 255.255.255.0
Router(dhcp-config)#exit
Router(config)#ip dhcp pool 20
Router(dhcp-config)#network 192.168.20.0 255.255.255.0
Router(dhcp-config)#exit

S1(config)#vlan 10
S1(config-vlan)#exit
S1(config)#
S1(config)#vlan 20
S1(config-vlan)#exit
S1(config)#interf range gi0/1-2
S1(config-if-range)#switchport mod trunk
S1(config-if-range)#exit
S1(config)#interf range fa0/1-10
S1(config-if-range)#swi mod acc
S1(config-if-range)#swi acc vlan 10
S1(config-if-range)#exit
S1(config)#interf rang fa0/11-20
```

```
S1(config-if-range)#swi mod acc
S1(config-if-range)#swi acc vlan 20
S1(config-if-range)#exit
S1(config)#ip dhcp snooping vlan 10,20
S1(config)#interf gi0/1
S1(config-if)#ip dhcp snooping trust
S1(config-if)#exit
# delete flash:vlan.dat
```

Az S2 pedig pont ugyan ez a konfiguráció (jelen esetben)
Az első 1005 vlan a vlan.dat-ba kerül, a többi pedig a running configba

VTP (virtuális trönkprotokoll)

VTP (virtuális trönkprotokoll) konfigurálása

Első módszer (switchportot is támogató routereken csak ez működik):

```
Switch# vlan database
Switch(vlan)# vtp domain tartománynév
```

Jelszó beállítása:

```
Switch(vlan)#vtp password jelszó
```

Protokoll verziójának beállítása:

```
Switch(vlan)# vtp v2-mode
```

Eszköz üzemmódjának beállítása
(alapesetben szerverként működik, a kliens csak fogadja a módosításokat, a transzparens átengedi a VTP-t és tőle függetlenül működtethet saját VLAN-okat):

```
Switch(vlan)# vtp mode server | client | transparent
```

Második módszer (globális konfig módban működik):

```
Switch(config)# vtp domain tartománynév
Switch(config)# vtp password jelszó
Switch(config)# vtp version 2
Switch(config)# vtp mode server | client | transparent
```

VTP ellenőrzése

```
Switch# show vtp status
Switch# show vtp password
```

VTP pruning

A kapcsolók nem továbbítják a trönk túlsó felére olyan VLAN-ok adatait, amikbe tartozó állomások nem léteznek a túloldalon, ezáltal kisebb lesz a fölösleges hálózati forgalom.

```
Switch(config)# vtp pruning
```

Forgalomirányítás (statikus)

IP útválasztás engedélyezése IPv4

```
R1(config)#ip routing
```

Statikus útvonalak IPv4

```
R1(config)#ip route 192.168.52.0 255.255.255.0 192.168.1.2 |
ser 0/0/0
```

Lebegő statikus útvonal IPv4

```
R1(config)#ip route 10.0.0.0 255.255.255.0 ser 0/0/0 150
```

```
// A végén a szám (metrika) segít súlyozni az útvonalat. Minél
kisebb ez a szám, annál kedvezőbb ez az útvonal.
```

Alapértelmezett út megadása IPv4

```
R1(config)# ip route 0.0.0.0 0.0.0.0 köv_ugrás ip címe | vagy
kiküldő interface
```

IP útválasztás engedélyezése IPv6

IPv6 statikus útvonal megadása:

```
R1(config)#ipv6 route 2001:470:1:1::/64 ser 0/0/0 | vagy
2001:470:1:2::1
```

IPv6 lebegő statikus útvonal megadása:

```
R1(config)#ipv6 route 2001:470:1:1::/64 ser 0/0/0 151
```

IPv6 alapértelmezett útvonal megadása:

```
R1(config)#ipv6 route ::/0 ser 0/0/0
```

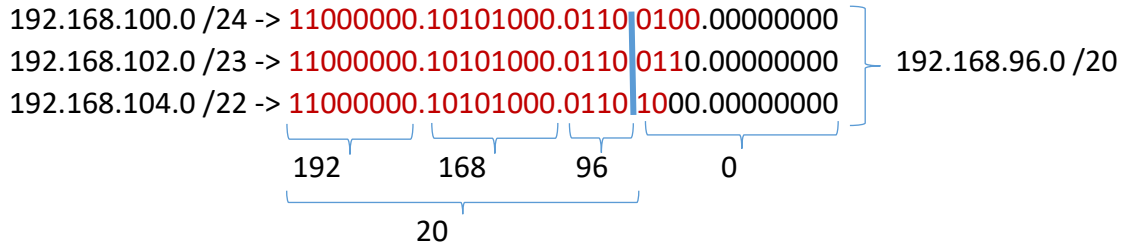
Routing tábla ellenőrzése

```
R1#show ip route
//IPV6 esetén
R1#show ipv6 route
```

Útvonalak összevonása

Amikor egy irányban van több cím, de hiány nélkül töltik ki a rendelkezésre álló tartományt, akkor összevonhatom őket az irányító táblában.

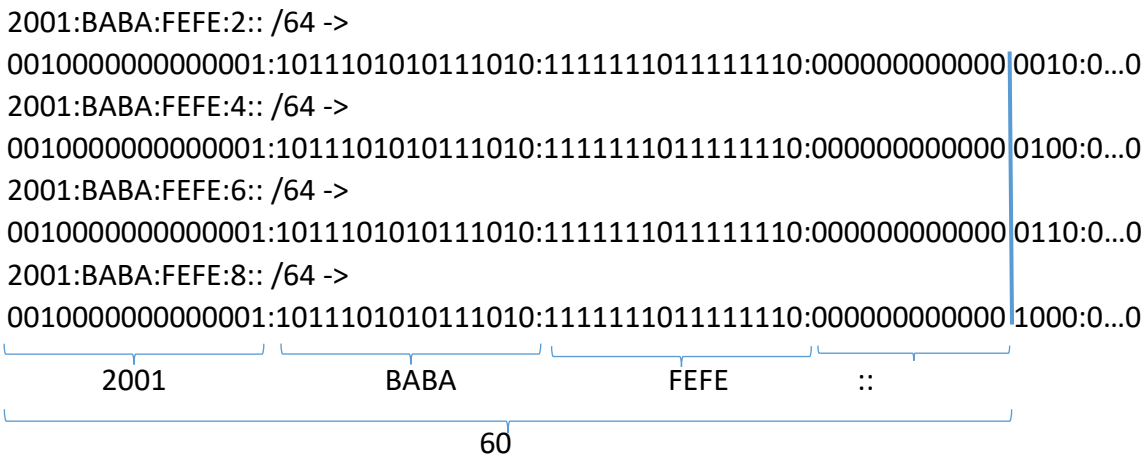
IPV4 címek esetén



Így a statikus útvonal megadása

```
R1(config)#ip route 192.168.96.0 255.255.240.0 ser 0/0/0
```

IPV6 címek esetén



Tehát 2001:BABA:FEFE:: /60

Így a statikus útvonal megadása

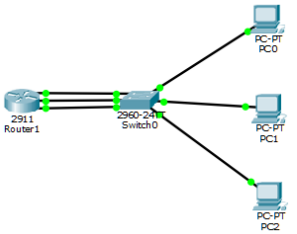
```
R1(config)#ip route 2001:BABA:FEFE:: /60 ser 0/0/0
```

Router on a stick

Amikor egy router több VLAN-ban is DHCP címeket oszt ki.

Egyszerű megvalósítás

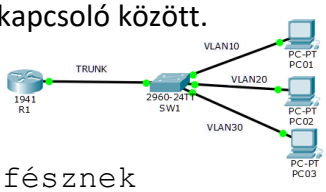
Amikor egy router annyi interfésszel van rákötve a hálózatra, amennyi VLAN-ban kell dolgoznia. Ez gazdaságtalan.



Ez esetben minden interfésznek adok egy saját ip címet. majd egy külön dhcp pool-t hozok létre. És már megy is.

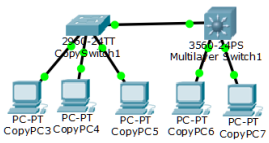
Trönk vonal segítségével

Ekkor máár csak egy trönk vonalat használok az útválasztó és a kapcsoló között.




```
Router(config)#interface gig0/0.1
//A végén a .1 lesz a neve ennek a subinterfésznek
Router(config-subif)#encapsulation dot1Q 10
//A végén a 10 rendeli hozzá a 10-es VLAN-hoz ezt a
subinterfészt
Router(config-subif)#ip address 172.16.0.1 255.255.0.0
Router(config-subif)#exit
Router(config)#interface gig0/0
Router(config)#no ip address
//Amikor felkapcsolom ezt az interfészt, az összes alinterfész
is felkapcsolódik
Router(config-if)#no shutdown
```

3. Rétegbeli kapcsolóval




```
Switch(config)#ip routing
//Ez irányítja majd a forgalmat a switchen belül
```

SVI létrehozása

```
Switch(config)#interface vlan /vlan id/
Switch(config-if)#ip add 192.168.100.1 255.255.255.0
Switch(config-if)#exit
```

VLAN létrehozása

```
Switch(config)#vlan /vlan id/
Switch(config-vlan)#name /vlan neve/
Switch(config-vlan)#exit
```

DHCP szolgáltatáés létrehozása

```
Switch(config)#ip dhcp pool /pool neve/
Switch(dhcp-config)#network 192.168.100.0 255.255.255.0
Switch(dhcp-config)#default-router 192.168.100.1
Switch(dhcp-config)#dns-server /dns ip címe/
Switch(dhcp-config)#exit
```

Szükséges TRUNK vonalak létrehozása

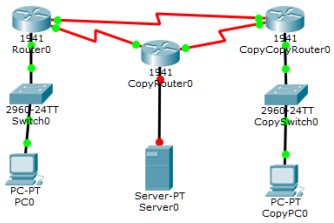
```
Switch(config)#interface range fa0/1-5
Switch(config-if-range)#switchport trunk encapsulation dot1q
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#exit
```

Hozzáférés a TRUNK vonalakhoz

```
Switch(config)#interface fa0/X
Switch(config-if)#switchport access vlan /vlan id/
Switch(config-if)#exit
```

Forgalomirányítás (dinamikus)

RIP (Routing Information Protocol)



- Távolságon alapuló útválasztás (egyszerűen megszámolja a hop-ok számát)
- Maximum 15 hop-ig jegyzi meg az útvonalakat
- Automatikusan frissíti a forgalomirányítási táblát. Időközönként (alapértelmezett: 30 másodperc) vagy ha változott a topológia
- Mindig a teljes routing táblát elküldi, azokba a hálózatokba ami be van állítva.

RIP v1

- Alapértelmezett beállítás
- Nincs subnet maszk támogatás
- Emiatt csak CLASSLESS hálózatokat (A, B, C címosztályokat) tud hirdetni
- Viszont kezeli a bejövő interfész maszkját, így folytonos alhálózatokat kezel.

RIP v2

- Jelszóval védhető (biztonságosabb)
- Több információ a csomagban (Adminisztratív távolság)
- Subnet maszk támogatás (VLSM vagy nem folytonos hálózatok támogatása)

IPV4 hálózaton

Engedélyezni a RIP-et:

```
Router(config)#router rip
```

Hozzárendelni egy hálózati tartományhoz:

```
Router(config-ruter)#network 192.168.0.0
```

Verziót állítani:

```
Router(config-router)#version 2
Router(config-if)#ip rip send version 2 //(küldés v2-ben)
Router(config-if)#ip rip receive version 2 //(fogadás v2-ben)
```

Hálózatok összevonásának tiltása (CIDR hirdetés):

```
Router(config-router)#no auto-summary
```

Hirdetési csomagok küldésének tiltása adott interfészre:

```
Router(config-router)#passive-interface FastEthernet 0/0
```

Statikus útvonal hirdetése:

```
Router(config-router)#redistribute static
```

Alapértelmezett útvonal hirdetése:

```
Router(config-router)#default-information originate
```

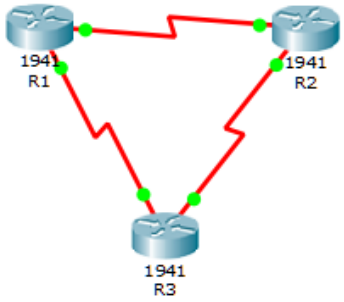
Beállítás ellenőrzése

```
Router# show ip protocols
//ellenőrizhető, hogy a RIP konfigurálva van
Router# show ip route
//megjeleníti az irányítótáblát, és így ellenőrizhető, hogy a
RIP szomszédoktól kapott útvonalak bekerültek a
forgalomirányító táblába
Router# debug ip rip
//használható a küldött és fogadott frissítésekben hirdetett
hálózatok megfigyelésére

Hitelesítés beállítása:
Router(config)#key chain Kulcs
Router(config-keychain)#key 1
Router(config-keychain-key)#key-string jelszo
Router(config)#int fa 0/0
Router(config-if)#ip rip authentication key-chain Kulcs
Router(config-if)#ip rip authentication mode md5

Látóhatár megosztás engedélyezése
Router(config-if)# ip split-horizon

RIP időzítők beállítása
Router(config-router)# timers basic 5 15 15 30

OSPF frissítés RIP frissítéssé alakítása
Router(config)#router rip
Router(config-router)#redistribute ospf 1 metric 3

EIGRP frissítés RIP frissítéssé alakítása
Router(config)#router rip
Router(config-router)#redistribute eigrp 100 metric 3
```

IPV6 hálózaton

```
RIPng protokoll:
Router(config)# ipv6 unicast-routing //RA üzenetek
Router(config)#int fa 0/0
Router(config-if)# ipv6 rip process1 enable //Csak az azonos
nevű hálózatokon érkeznek frissítések
Router (config-if)# ipv6 rip process1 default-information
originate //alapértelmezett átjáró hírdetése zen az
interfészen
Router (config)# ipv6 router rip process1
Router (config)#ipv6 route ::/0 serial 0/0/0 //végsű utvomal
valamint
Router# show ipv6 protocols
Router# show ipv6 route
```

OSPF(Open Shortest Path First /Legrövidebb utat először)



Open Shortest Path First - RFC 1131
Az Open Shortest Path First (OSPF) jellemzői:

- Link állapot alapú IGP protokoll.
- Új, 90'-es évektől alapértelmezettként javasolt.
- AS-nél kisebb hálózati egység, terület (area) használata.
- Forgalomirányítók (nem diszjunkt) osztályozása:
  - Területen belül működő forgalomirányítók.
  - Területek határán álló forgalomirányítók.
  - Gerinchálózaton (backbone) üzemelő forgalomirányítók.
  - AS határon működő forgalomirányítók.
- Egyenlő költségű többutas irányítás lehetősége.
- IP fejléc „Szolgáltatás típusa" mezőjének használata.
- Mai verzió: OSPF V2 (RFC 1583).

OSPF területek
A döntési folyamat (Dijkstra algoritmus) alapja a terület (area).
A területek „csillag alakzatot" formáznak, középpontjában a területeket összekötő speciális területtel (backbone).
A terület határ router-ek feladata összetett:

- Minden területhez (külön) döntési folyamat.
- A területekből tanult információk összegzése.
- Az összegzett információk bevitele a többi területbe.

Területek közötti forgalomirányítás (inter area routing):

- Routing a forrás területben a határ router-ig.
- Routing a backbone-on a cél terület határ router-ig.
- Routing a cél területben a cél hálózatig.

OSPF – speciális fogalmak
Designated Router

- Olyan router, mely egy LAN nevében propagál link-állapot (LSA) információkat.

Pszeudonode

- Egy üzenetszórásos alhálózatban maga az alhálózat egy ál csomópontnak (pszeudonode) tekinthető. A designated IS a pszeudonode nevében propagálja az LS információkat.
  (A szükséges információcsere száma n*2* nagyságrendről 2n nagyságrendre csökkenthető)

OSPF adatok nyilvántartása
Az OSPF router táblázatának legfontosabb elemei:

- Cél típusa (hálózat, terület határ router, AS határ router).
- Cél azonosító (IP szám).
- Szolgáltatás típusa.
- A célhoz vezető út/utak megadása:
  - Út típusa (itra-area, inter-area, AS-external).

- Út költsége.
- Következő forgalomirányító (IP szám, elérés interfésze).

```
R1(config)#router ospf 115 /területet ide lehet írni/ area 0
//a 115-nek csak a routeren belól van jelentőssége
R1(config-router)#log-adjacency-changes
R1(config-router)#network 195.220.123.0 0.0.0.255 area 0
R1(config-router)#exit
```

Minden oktet 255–ből kivonva

```
//a terület 0 az azonos OSPF területeket jelenti, ahol a
kommunikáció megtörténik
```

Staitkus ótvonalak hírdetése

```
Router(config-router)#redistribute static
```

Alapértelmezett átjáró hírdetése

```
Router(config-router)#default-information originate
```

Router-azonosító megadása:

```
R1(config-router)#router-id 200.0.0.1
```

Soros összeköttetés sávszélességének megadása (kbit/s):

```
R1(config)#interface fa0/1
R1(config-if)#bandwith 115000
```

Interfész prioritásának megadása (ha 0, nem vesz részt a DR/BDR választásban):

```
R1(config)#interface fa0/1
R1(config-if)#ip ospf priority 100
```

Loopback interfész létrehozása:

```
(config)#interface loopback 0
(config-if)#ip address 200.0.0.1 255.255.255.255
```

Passzív interfész létrehozása:

```
R1(config-router)#passive-interface default
R1(config-router)#{no} passive-interface Serial 0/0/0
```

Költségérték módosítása:

```
R1(config)#interface fa0/1
R1(config-if)#ip ospf cost 100 (az érték 1-255 lehet)
```

Hitelesítés jelszóval:

```
R1(config-router)#area 0 authentication
R1(config-if)#ip ospf authentication-key titok
```

Hitelesítés MD5 segítségével:

```
R1(config-router)#area 0 authentication message-digest
```

```
R1(config-if)#ip ospf message-digest-key 1 md5 titok123
```

Ellenőrzés:

```
R1#sh ip ospf interface
R1#sh ip ospf neighbour [detail]
R1#debug ip ospf adj | events
R1#show ip ospf database
R1#show ip route ospf
R1# show ip protocols
```

Hello és halott időzítők beállítása:

```
R1(config)#interface fa0/1
R1(config-if)#ip ospf hello-interval 15
R1(config-if)#ip ospf dead-interval 50
```

Alapértelmezett útvonal hirdetése:

```
R1(config-router)#default-information originate
```

Összevont útvonal konfigurálása:

```
R1(config-router)#area terület-azonosító range IP-cím maszk
```

Referencia-sávszélesség értékének módosítása:

```
R1(config-router)#auto-cost reference-bandwidth
```

A módosítások érvénybe léptetése:

```
R1(config-router)#clear ip ospf process
```

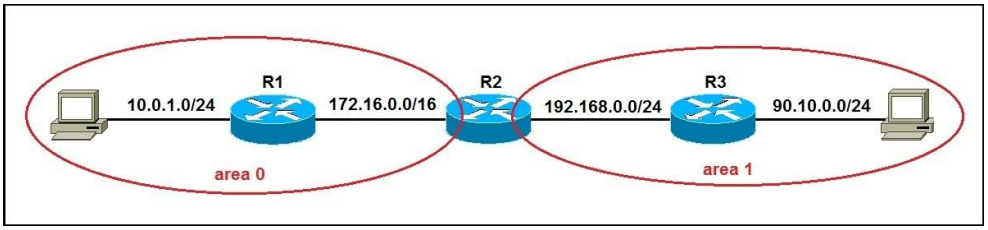
RIP frissítés OSPF frissítéssé alakítása:

```
R1(config-router)#redistribute rip subnets
```

EIGRP frissítés OSPF frissítéssé alakítása:

```
R1(config)#router ospf 1
R1(config-router)#redistribute eigrp 10 subnets
```

Több területű OSPF




```
R1(config)#router ospf 1
R1(config-router)#network 10.0.1.0 0.0.0.255 area 0
R1(config-router)#network 172.16.0.0 0.0.255.255 area 0
R1(config-router)#router-id 1.1.1.1
```

```
R3(config)#router ospf 1
R3(config-router)#network 192.168.0.0 0.0.0.255 area 1
R3(config-router)#network 90.10.0.0 0.0.0.255 area 1
R3(config-router)#router-id 3.3.3.3
```

```
R2(config)#router ospf 1
R2(config-router)#network 172.16.0.0 0.0.255.255 area 0
R2(config-router)#network 192.168.0.0 0.0.0.255 area 1
R2(config-router)#router-id 2.2.2.2
```

OSPFv3 (IPV6) protokoll alapbeállítása

```
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1 //ez nem ip cím, hanem
process azonosító
R1(config-rtr)#exit
R1(config)#int fa 0/1
R1(config-if)#ipv6 ospf 1 area 0
```

Ellenőrzés:

```
R1#sh ipv6 ospf interface
R1#sh ipv6 ospf neighbour [detail]
R1#debug ipv6 ospf adj | events
R1#show ipv6 ospf database
R1#show ipv6 route ospf
R1# show ipv6 protocols
```

OSPF autentikáció

EIGRP protokoll

Alapbeállítás:

```
R1(config)#router eigrp 111 //ez a szám ugyanaz kell, hogy legyen minden routeren
R1(config-router)#no auto-summary
R1(config-router)#network 192.168.1.0 //maszk nélkül – osztályos címként kezeli majd
R1(config-router)#network 200.0.0.0 255.255.255.252 //rendes maszkkal – de meg fogja fordítani beírás után
R1(config-router)#network 201.1.1.0 0.0.0.3 //fordított maszkkal
```

Alapértelmezett útvonal és a statikus ótvonalak hirdetése:

```
R1(config-router)#redistribute static
```

Passzív interfész beállítása

```
R1(config-router)#passive-interface Fa 0/0
```

Nem egyenlő költségű útvonalakon való terheléselosztás:

```
R1(config-router)#variance 5
```

(ekkor a legjobb útvonalnál 5-ször rosszabb költségű útvonalakat is bevonja az irányítótáblába) Közvetlenül kapcsolódó hálózatok bevonása az irányítási folyamatba (ezekbe nem küld EIGRP csomagokat):

```
R1(config-router)#redistribute connected
```

A szomszédsági viszonyok változásainak követése

```
R1(config-router )#eigrp log-neighbor-changes
```

Soros összeköttetések sávszélessége

```
R1(config-if)#bandwith 1544
```

Hello időzítő értékének módosítása (default: T1< and NBMA = 60s T1> = 5s)

```
R1(config-if)#ip hello-interval eigrp 1 10
```

Halott időzítő értékének módosítása (default: T1< and NBMA = 180s T1> = 15s)

```
R1(config-if)#ip hold-time eigrp 1 10
```

Útvonalösszevonás:

```
R1(config-if)#ip summary-address eigrp 111 192.168.0.0 255.255.0.0
```

Hitelesítés beállítása:

```
R1(config)#key chain Kulcs
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string jelszo
R1(config)#int fa 0/0
R1(config-if)#ip authentication key-chain eigrp 1 Kulcs
```

```
R1(config-if)#ip authentication mode eigrp 1 md5
```

EIGRP mérték

EIGRP alapértelmezetten a következő értékeket használj a legjobb útvonal számításhoz:

- sávszélesség
- késleltetés

Egyéb értékek:

- megbízhatóság
- terhelés

A számításhoz használt súlyozások k1, k2, k3, k4, k5 azonosítóval kerülnek kijelölésre. Minden k értéknek 0 vagy 1 értéke van. Ha 0 értéke van, nem vesszük figyelembe.

- k1 - sávszélesség - alapértelmezetten van
- k2 - terhelés
- k3 - késleltetés - alapértelmezetten van
- k4 - megbízhatóság
- k5 - megbízhatóság

A mérték számítási módjának egyeznie kell a szomszédoknál.
R1(config-router)# metric weights tos k1 k2 k3 k4 k5

Ellenőrzés:

```
R1# show ip protocols
```

Képlet:

metric = [K1*bandwidth+ (K2*bandwidth/(256-load)+K3*delay]*[K5/(reliability + K4)]

Interfészek értékei

```
R1# show interfaces Serial 0/0/0
R1# show interfaces GigabitEthernet 0/0
```

- BW - interfész sávszélesség
- DLY - interfész késleltetése (mikroszekundum)

Sávszélesség állítás:

```
R1(config)# interface s 0/0/0
R1(config-if)# bandwidth 64
```

```
R2(config)# interface s 0/0/0
R2(config-if)# bandwidth 64
R2(config-if)# interface s 0/0/1
R1(config-if)# bandwidth 1024

R3(config)# interface s 0/0/1
R3(config-if)# bandwidth 1024
```

A k értékek beállítása:
Ellenőrző parancsok:

```
R1#show ip eigrp neighbors
R1#show ip eigrp topology [all-links]
R1#debug eigrp fsm | packets
```

RIP frissítés EIGRP frissítéssé alakítása
sávszélesség|késleltetés|megbízhatóság|Terhelés|MTU]

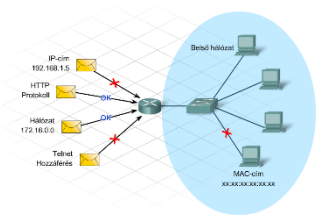
```
R1(config)#router eigrp 100
R1(config-router)#redistribute rip metric 128 1000 100 100 100
```

OSPF frissítés EIGRP frissítéssé alakítása
[sávszélesség|késleltetés|megbízhatóság|Terhelés|MTU]

```
R1(config)#router eigrp 100
R1(config-router)#redistribute ospf 1 metric 128 1000 100 100
100
```

Forgalom szűrés

Ajánlott a listát először szövegszerkesztőben elkészíteni és csak a kész listát bemásolni mert a későbbi javítás így megkönnyíthető!



Hozzáférési (ACL, Access Control List) listák megadása

IPV4

Normál ACL szintaktika:

```
R1(config)#access-list szám permit|deny host_ip|ip_tartomány
wildcard maszkja
```

Normál ACL a 193.225.10.0/24 célhálózathoz enged:

```
R1(config)#access-list 1 permit 193.225.10.0 0.0.0.255
```

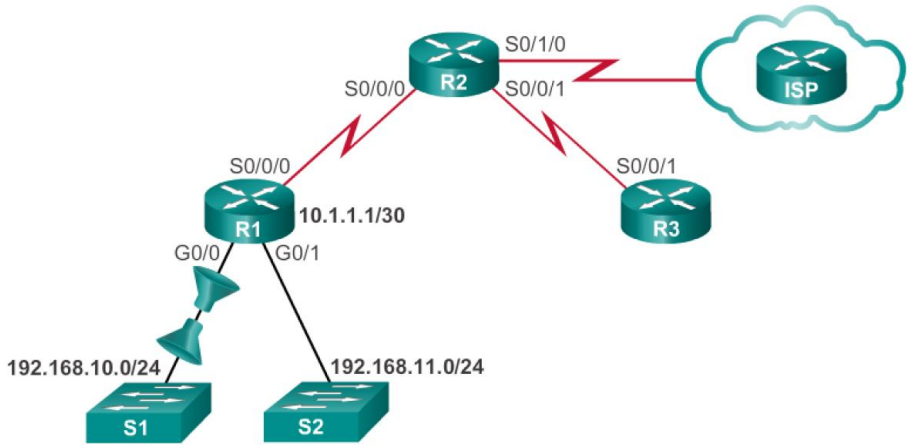
Normál ACL egy számítógép tiltásához:

```
R1(config)#access-list 1 deny host 195.140.100.5
```

established //lehetővé teszi, hogy csak a forrásról indított kérdésre érkezhet válasz

HTTP szűrése

- A HTTP működése megköveteli a visszafelé irányuló forgalom engedélyezését a webszervertől a kapcsolatot kezdeményező kliens felé.
- A hálózati rendszergazda engedélyezni kívánja ezt a visszatérő HTTP-válaszforgalmat, de minden mást tiltani akar.
- Az **established** paraméter csak azokat a csomagokat engedi visszatérni, amelyek válaszok egy hálózatból indított kérésre.
- Egyezés akkor történik, ha a visszajövő TCP-szegmens tartalmaz ACK vagy RST (reset) bitet, amely mutatja, hogy a csomag egy már létező kapcsolathoz tartozik




```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255
any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255
any eq 443
```

```
R1(config)#access-list 104 permit tcp any 192.168.10.0
0.0.0.255 established
R1(config)#interface gig0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip access-group 104 out
```

A következő példában tiltjuk a 195.220.0.0/16 hálózat felől a HTTP (80-as port) kéréseket bármilyen célhálózat felé:

```
R1(config)#access-list 101 deny tcp 195.220.0.0 0.0.255.255
0.0.0.0 0.0.0.0 eq 80
```

Kierjesztett/extended nevesített ACL használatával

1. lépés Globális konfigurációs módban
    - ip access-list extended név parancs a kiterjesztett lista nevének megadása.
2. lépés Nevesített konfigurációs módban
    - adjuk meg a szabályozó feltételeket a permit vagy a deny kulcsszó után.

Kiterjesztett ACL szintaktikája:

```
R1(config)#access-list szám permit|deny protokoll forrás_ip
reverse-maszk cél_ip reverse-maszk
[eq port [established]]
```

```
R1(config)#ip access-list extended SURFING
R1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq
80
R1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq
443
R1(config-ext-nacl)#exit
R1(config)#ip access-list extended BROWSING
R1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255
established
R1(config-ext-nacl)#exit
R1(config)#interface gig0/0
R1(config-if)#ip access-group SURFING in
R1(config-if)#ip access-group BROWSING out
R1(config-if)#exit
```

Portok megadásához használhatók:

- **eq** ha egy portot adunk meg (equal)
- **ne** ha nem azt a portot akarjuk (not equal)
- **lt** ha megadott portnál kisebbeket akarjuk
- **gt** ha megadott portnál nagyobbakat akarjuk
- **range x to y** ha portszámok tartományát akarjuk

Nevesített ACL:

```
R1(config)#ip access-list standard ACL-IN
R1(config)#ip access-list extended ACL-OUT
R1(config-ext-nacl)#permit icmp any any
```

Az ACL definiálása után az ACL-t interfészhez kell rendelni. Fontos megadni, hogy kimenő vagy bejövő interfészhez rendeljük-e!

```
R1(config)#interface Serial 0/0/0
R1(config-if)#ip access-group 1 out
```

ACL leírás megadása:

```
R1(config)#access-list 1 remark //ez tilt mindent
```

WEB kiszolgáló engedélyezése:

```
R1(config)#ip http server
R1(config)#ip http secure-server
R1(config)#ip http authentication local
```

IPV6

Csak nevesített ACL-ek lehetnek!

Eltérések az IPv4 és IPv6 ACL között

- IPv6 ACL alkalmazása
  - ip access-group parancs helyett az ipv6 trafficfilter

paranccsal rendeljük interfészhez az ACL-t.

- Nincsenek helyettesítő maszkok
  - Ehelyett az előtag-hossz (prefix-lenght) határozza meg, hogy az IPv6 cél- vagy forráscímből mekkorarész kerüljön vizsgálatra.
- További alapértelmezett utasítások
  - Két további, implicit "permit" utasítás került minden IPv6 hozzáférési lista végére. Az IPv6 is tartalmaz egy hasonló **deny ipv6 any any** utasítást minden IPv6 ACL végén.
    - **permit icmp any any nd-na**
    - **permit icmp any any nd-ns**
  - Ez a két utasítás teszi lehetővé a forgalomirányító számára, hogy részt vegyen z IPv4 ARP-jéhez hasonló IPv6-címfeloldásban.

IPv6 ACL konfigurálásának három lépése

- **1. lépés** Globális konfigurációs módban
  - **`ipv6 access-list`** név paranccsal az ACL létrehozása
- **2. lépés** Nevesített ACL konfigurációs módban
  - a permit vagy a **`deny`** utasításokkal a feltételek megadása.
- **3. lépés** Térjünk vissza privilegizált EXEC módba az **`end`** paranccsal.

```
R1(config)#ipv6 access-list lista_neve

R1(config-ipv6-acl)# deny | permit protocol
{ source-ipv6-prefix / prefix-length | any | host source-ipv6-
```

address } [*operator* [*port-number*]] { *destination-ipv6-prefix/ prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*]

R1(config-ipv6-acl)# deny | **permit tcp** { *source-ipv6-prefix* **/** *prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] { *destination-ipv6- prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**ack**] [**dscp** *value*] [**established**] [**fin**] [**log**] [**log-input**] [**neq** { *port* | *protocol* }] [**psh**] [**range** { *port* | *protocol* }] [**rst**] [**sequence** *value*] [**syn**] [**time-range** *name*] [**urg**]

R1(config-ipv6-acl)# deny | **permit udp** { *source-ipv6-prefix* **/** *prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dscp** *value*] **[log**] [log-input] [**neq** { *port* | *protocol* }] [**range** { *port* | *protocol* }] [**sequence** *value*] [**time-range** *name*]

R1(config-ipv6-acl)# deny | **permit icmp** { *source-ipv6-prefix* **/** *prefix-length* | **any** | **host** *source-ipv6-address* } [*operator* [*port-number*]] { *destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [*icmp-type* [*icmp-code*] | *icmp-message*] [**dscp** *value*] [**log**] [log-input] [**sequence** *value*] [**time-range** *name*]

Példa:

```
R1(config)# ipv6 access-list CISCO
R1(config-ipv6-acl)# deny tcp any any gt 5000
R1 config-ipv6-acl)# deny ::/0 lt 5000 ::/0 log
R1(config-ipv6-acl)# permit icmp any any
R1(config-ipv6-acl)# permit any any
```

Majd Interface-re kell tenni:

```
R1(config)# interface gigabitethernet1/0/3
R1(config-if)# ipv6 address 2001::/64 eui-64
R1(config-if)# ipv6 traffic-filter CISCO {in | out}
```

Switch esetén:

```
Switch(config)# interface gigabitethernet1/0/3
Switch(config-if)# no switchport
Switch(config-if)# ipv6 address 2001::/64 eui-64
Switch(config-if)# ipv6 traffic-filter CISCO {in | out}
```

Ellenőrzése:

```
show access-lists
show ipv6 access-list [ access-list-name ]
show ipv6 interface interfész
show running-config
```

SSH belépések korlátozása

Ezt kivételesen mindig a célhoz a legközelebb kell tenni befelé irányban.

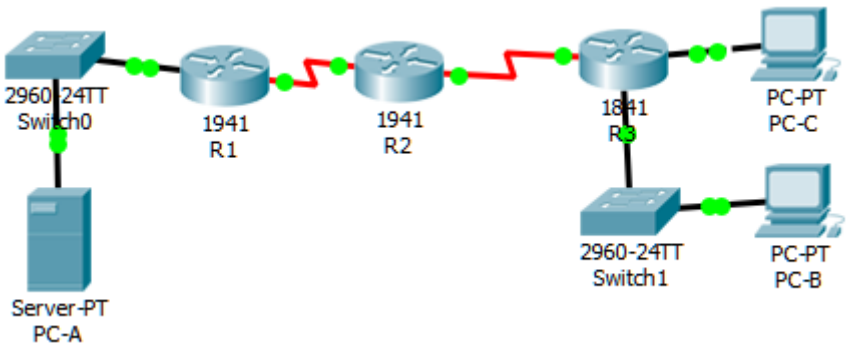
Melyik hálózatból engedem (vagy tiltom)

```
access-list 1 permit 10.10.100.192 0.0.0.63
```

Beállítom a VTY sávon belüli vonalon

```
line vty 0 15
access-class 1 in
login local
transport input ssh
```

Zóna alapú tűzfal beállítása



ZBPF config

1.) zónák létrehozása
   a. …(config)# zone security EZANEVE1
   b. …(config)# zone security EZANEVE2
   c. …(config)# zone security EZANEVE3

2.) forgalom osztályozása
   a. …(config)# class-map type inspect match-any | match-all FORGTIP1
   b. …(config-cmap)# match protocol PROTTIPUS1
   c. …(config-cmap)# match protocol PROTTIPUS2

3.) tevékenység megadása
   a. …(config)# policy-map type inspect TEVNEVE1
   b. …(config-pmap)# class type inspect FORGTIP1
   c. …(config-pmap-c)# inspect | drop | pass

4.) zónapárok kijelölése
   a. …(config)# zone-pair security ZONAPARNEVE source EZANEVE1 destination EZANEVE3
   b. …(config-sec-zone-pair)# service-policy type inspect TEVNEVE1

5.) interfészek zónába helyezése
   a. …(config-if)# zone-member security EZANEVE1

ALAPBEÁLLÍTÁSOK

```
security passwords min-length 10
ip domain-name ccnasecurity.com
crypto key generate rsa
1024
username admin01 secret cisco12345
line console 0
login local
exec-timeout 5 0
logging synchronous
exit
line aux 0
login local
exec-timeout 5 0
exit
line vty 0 4
login local
```

```
transport input ssh
exec-timeout 5 0
enable secret class12345
do write
```

Létrehozom a zónákat

```
R3(config)# zone security zóna-neve
R3(config)# zone security INSIDE
R3(config)# zone security CONFROOM
R3(config)# zone security INTERNET
```

Megadom, hogy miket kell vizsgálni:

match-any : **VAGY KAPCSOLAT**
match-all : **ÉS KAPCSOLAT**

```
R3(config)# class-map type inspect match-any gyüjtemény-neve
R3(config)# class-map type inspect match-any INSIDE_PROTOCOLS
R3(config-cmap)# match protocol vagy ip cím vagy egyéb
R3(config-cmap)# match protocol tcp
R3(config-cmap)# match protocol udp
R3(config-cmap)# match protocol icmp

R3(config)# class-map type inspect match-any
CONFROOM_PROTOCOLS
R3(config-cmap)# match protocol http
R3(config-cmap)# match protocol https
R3(config-cmap)# match protocol dns
```

Létrehozom a szabályt, amely magába foglalja, hogy melyik szempontokat kell nézni.
Majd megadom neki, hogy ezek alapján vizsgálja (Inspect-vizsgálat, pass-ne tegyen semmit, drop-ne engedjen semmit)

```
R3(config)# policy-map type inspect (neve)-INSIDE_TO_INTERNET
R3(config-pmap)# class type inspect (neve)-INSIDE_PROTOCOLS
R3(config-pmap-c)# inspect (vizsgálat módja -inspect, drop,
pass)

R3(config)# policy-map type inspect CONFROOM_TO_INTERNET
R3(config-pmap)# class type inspect CONFROOM_PROTOCOLS
R3(config-pmap-c)# inspect
```

Melyik zónák hogyan komunikáljanak egymással
létrehozom a zóna párokat és műveletet:

```
R3(config)#zone-pair security (zonapár neve)-
INSIDE_TO_INTERNET source (honnan-neve)INSIDE destination
(hova-neve)INTERNET
```

```
R3(config-sec-zone-pair)#service-policy type
(tevékenység)inspect (neve)INSIDE_TO_INTERNET
R3(config-sec-zone-pair)#exit
R3(config)#zone-pair security CONFROOM_TO_INTERNET source
CONFROOM destination INTERNET
R3(config-sec-zone-pair)#service-policy type inspect
CONFROOM_TO_INTERNET
R3(config-sec-zone-pair)#exit
R3(config)#
```

Majd rá teszem az interfészre:

```
R3(config)# interface fa0/0
R3(config-if)# zone-member security (zona neve)-CONFROOM
R3(config)# interface fa0/1
R3(config-if)# zone-member security INSIDE
R3(config)# interface s0/0/1
R3(config-if)# zone-member security INTERNET
```

ellenőrzés:

```
show zone-pair security
show policy-map type inspect zone-pair
```

DHCP beállítása

IPV4 esetén

```
Router(config)# ip dhcp pool pool1
Router(dhcp-config)# network 172.16.0.0 /16
//Érdemes ezt útojára beállítani, mert a parancs kiadása
után már egybül osztja a cím eket
Router(dhcp-config)# domain-name cisco.com
Router(dhcp-config)# dns-server 172.16.1.103 172.16.2.103
Router(dhcp-config)# default-router 172.16.1.100
172.16.1.101
//Ez az alapértemlezett átjáró. A második cím nem kötelező
Router(dhcp-config)# lease 30 //a maximálisan kioszthatő
címek
Router (config-dhcp)#lease 1 12 30 //nap óra perc formátum
Router (config-dhcp)#option 150 ip 192.168.0.1
//IP  telefonnál a tftp szerver címe, innen jön a config

Router(config)# ip dhcp excluded-address 172.16.1.100
172.16.1.199
//Ezek a kizárások mindig érvénben maradnak. Kikapcsolni a
no ip dhcp excluded-address 172.16.1.100 172.16.1.199
paranccsal leht. Egyszerre több különböző kizárás is
megadható.
```

```
Switch(config)#ip default-gateway 10.0.0.254
//alapértemlezett átjáró switch esetén
```

Intewrfész ip címe dhcp-vel

```
Router (config)#interface fastethernet0/0
Router (config-if)#ip  address  dhcp
Router (config-if)#no shutdown
```

IP cím kötése MAC címhez

```
Router(config)#ip dhcp pool FIXIP
Router(dhcp-config)#host 200.20.2.20 255.255.255.0
Router(dhcp-config)#hardware-address 01b7.0813.8811.66
```

vagy

```
ip dhcp pool pool1
 host 172.16.5.10
 client-identifier  0002.0528.f405.23
 client-name testnev
```

Külső DHCP szerver

Ha a DHCP szerver másik hálózati szegmensen van, akkor a DHCP DISCOVER-t fogadó interfészen meg kell adni a DHCP szerver címét:

```
R1(config-if)# ip helper-address 192.168.10.1
```

Majd ezen a szerveren létre kell hozni egy ebbe a hálózatba tartozó dhcp pool-t.

IPV6 (dhcp) SLAAC segítségével

```
Router(config)#ipv6 unicast-routing /RA üzenetek küldésének
bekapcsolása
//Igy már tud IPV6 dhcp serverként működni a router
– RA üzenetek küld
R1(config)#int fa 0/0
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 address 2001:470:1:1::1/64
R1(config-if)#no shutdown

vagy:
 R1(config)#ipv6 unicast-routing
 R1(config)#int fa 0/0
 R1(config-if)#ipv6 enable
 R1(config-if)#ipv6 address 2001:db8:1111:2::/64 eui-64
 //Ez esetben saját magának állít be IPV6 címet a MAC cím
 segítségével
 R1(config-if)#no shutdown
 vagy:
 R1(config)#ipv6 unicast-routing
 R1(config)#int fa 0/0
 R1(config-if)#ipv6 enable
 R1(config-if)#ipv6 address dhcp | autoconfig
 R1(config-if)#no shutdown
```

Állapotmentes DHCPv6 szerver konfigurálás

• 1. lépés: Az IPv6-irányítás engedélyezése
– Az ipv6 unicast-routing paranccsal engedélyezhető az IPv6-irányítás. (ICMPv6 RA-üzeneteinek küldéséhez).
• 2. lépés: DHCPv6 készlet beállítása
– Az ipv6 dhcp pool készlet-neve paranccsal.
• 3. lépés: A készlet paramétereinek beállítása
– Pl: DNS-szerver címe, valamint a tartomány neve.
• 4. lépés: A DHCPv6-interfész beállítása
– ipv6 dhcp server készlet-neve
– ipv6 nd other-config-flag

```
R1(config)# ipv6 unicast-routing //ezek után már képes osztani
IPV6 címeket, amelyek a MAC cím segítségével készít el
R1(config)#ipv6 dhcp pool STATLESS
R1(config-dhcpv6)#dns-server [ipv6 address]
R1(config-dhcpv6)#domain-name [name]
R1(config-dhcpv6)#exit
R1(config)#interface gig0/0
R1(config-if)#ipv6 dhcp server STATELESS
R1(config-if)# ipv6 address baba::1/64 //igy kap egy fix címet
```

```
 R1(config-if)#no ipv6 nd managed-config-flag //emiatt NEM
állapottartó
R1(config-if)#ipv6 nd other-config-flag //emiatt állapotmentes
```

Routeren, ha nem adok neki fix címet

```
R1(config)#interface gig0/0
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 adress autoconfig
```

Állapottartó DHCPv6 server konfigurálása

```
R1(config)# ipv6 unicast-routing
R1(config)#ipv6 dhcp pool STATLEFUL
R1(config-dhcpv6)#address prefix 2001:db8:cafe:1::1/64
lifetime infinite infinite
R1(config-dhcpv6)#dns-server [ipv6 address]
R1(config-dhcpv6)#domain-name [name]
R1(config-dhcpv6)#exit

R1(config)#interface gig0/0
R1(config-if)#ipv6 dhcp server STATEFUL
R1(config-if)#ipv6 address 2001:db8:cafe:1::1/64
R1(config-if)#ipv6 nd managed-config-flag //emiatt
állapottartó
```

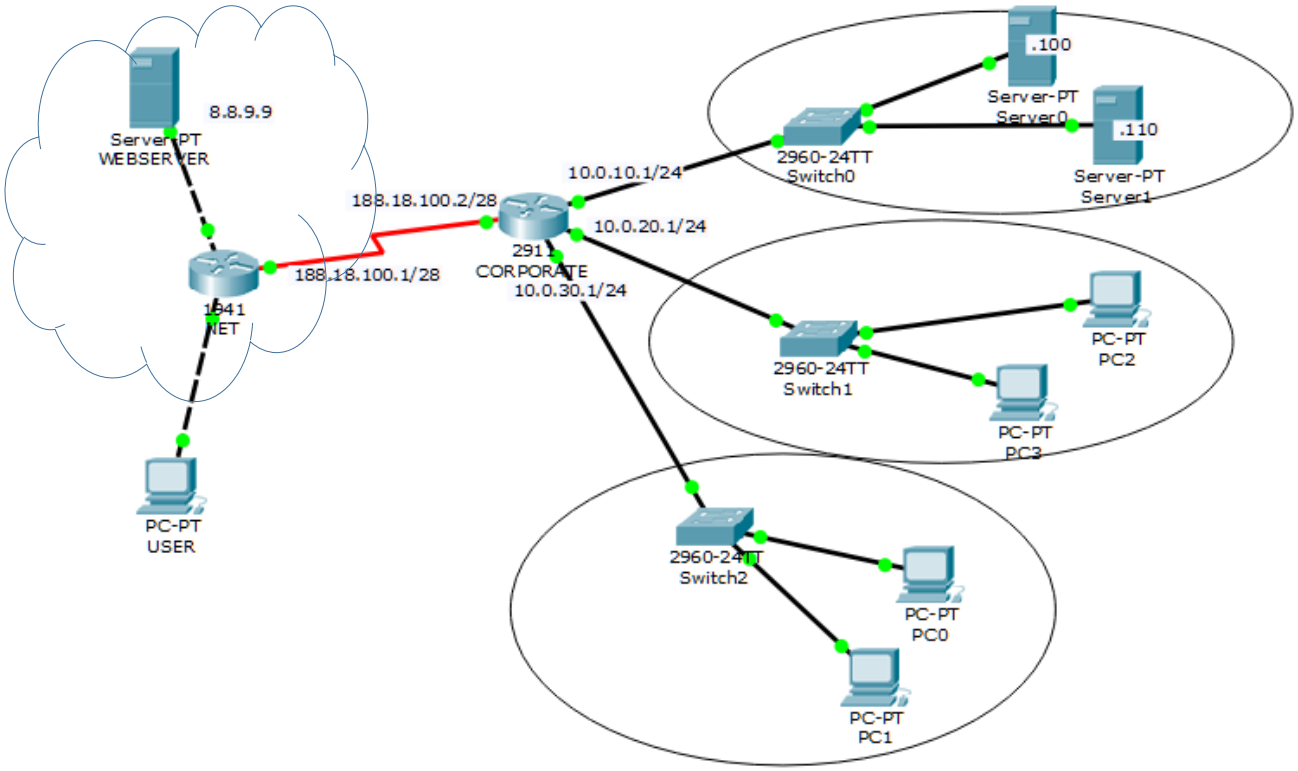
Ellenőrzés

```
show ipv6 dhcp pool
show running-config
```

Kliens ellenőrzése

```
show ipv6 interface
debug ipv6 dhcp detail
```

NAT beállítása



PAT BEÁLLÍTÁSA

- Csak egy külsö ip cím van -> kell inerface.
- Engedni kell a belső címeket
- Megadni a külső interface-t és megadni a belső interface-ket
- Majd összerendelni a belső engedélyezett címeket a külső interface-val + overload

Ha csak a második hálózatnak kell tudni netezni

PAT BEÁLLÍTÁSA

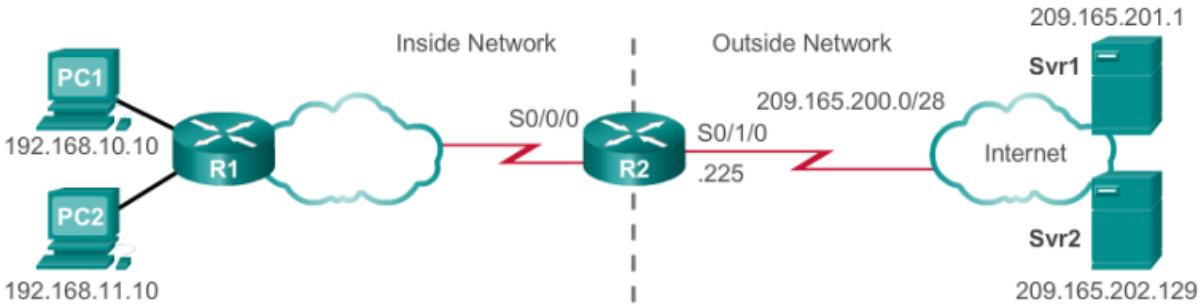
```
CORPORATE>enab
CORPORATE#conf t
CORPORATE(config)#access-list 1 permit 10.0.20.0 0.255.255.255
```

ezzel állítom be, hogy mi történjen a Pat során

```
CORPORATE(config)#interf gig0/1
CORPORATE(config-if)#ip nat inside
CORPORATE(config-if)#exit
CORPORATE(config)#interf ser0/0/0
CORPORATE(config-if)#ip nat outside
CORPORATE(config-if)#exit
CORPORATE(config)#ip nat inside source list 1 interface serial
0/0/0 overload
```

Alapértelmezett útvonal megadása a külvilág eléréséhez:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 ser 0/0/0
```



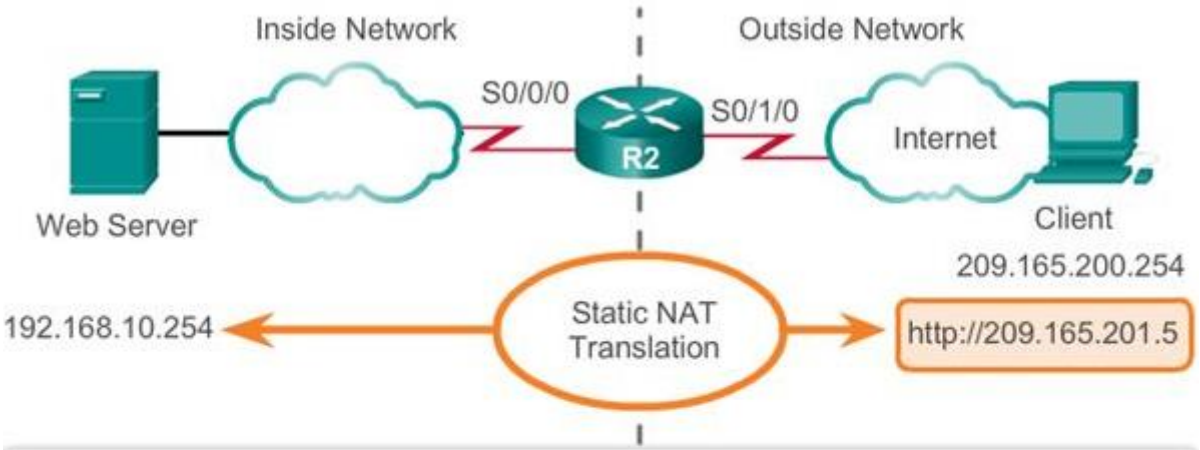
Define a pool of public IPv4 addresses under the pool name NAT-POOL2.
R2(config)# **ip nat pool NAT-POOL2 209.165.200.226 209.165.200.240 netmask 255.255.255.224**
Define which addresses are eligible to be translated.
R2(config)# **access-list 1 permit 192.168.0.0 0.0.255.255**
Bind NAT-POOL2 with ACL 1.
R2(config)# **ip nat inside source list 1 pool NAT-POOL2 overload**

Identify interface serial 0/0/0 as an inside NAT interface.
R2(config)# **interface Serial0/0/0**
R2(config-if)# **ip nat inside**

Identify interface serial 0/1/0 as the outside NAT interface.
R2(config)# **interface Serial0/1/0**
R2(config-if)# **ip nat outside**

STATIKUS NAT BEÁLLÍTÁSA

- megadom a külső interce-t
- megadom a belső interface-t
- majd összerendelem




```
Establishes static translation between an inside local address and
an inside global address.
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5

R2(config)# interface Serial0/0/0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
Identifies interface serial 0/0/0 as an inside NAT interface.
R2(config-if)# ip nat inside
R2(config-if)# exit

R2(config)# interface Serial0/1/0
R2(config-if)# ip address 209.165.200.225 255.255.255.224
Identifies interface serial 0/1/0 as the outside NAT interface.
R2(config-if)# ip nat outside
```

Ellenőrzés:

CORPORATE#show ip nat translations

```
Pro  Inside global     Inside local       Outside local      Outside global
---  188.18.100.14     10.0.10.100        ---                ---
tcp 188.18.100.14:80   10.0.10.100:80     195.180.13.50:1027
195.180.13.50:1027
tcp 188.18.100.14:80   10.0.10.100:80     195.180.13.50:1030
195.180.13.50:1030
tcp 188.18.100.2:1025  10.0.20.2:1025     8.8.9.9:80         8.8.9.9:80
```

CORPORATE#show ip nat statistics – clear ip nat statistics

```
Total translations: 4 (1 static, 3 dynamic, 3 extended)
Outside Interfaces: Serial0/0/0
Inside Interfaces: GigabitEthernet0/0 , GigabitEthernet0/1
Hits: 23  Misses: 3
Expired translations: 0
Dynamic mappings:
```

Alapértelmezett útvonal megadása a külvilág eléréséhez:
R1(config)#ip route 0.0.0.0 0.0.0.0 ser 0/0/0

DINAMIKUS NAT BEÁLLÍTÁSA

- megadom a külső interf
- megadom a belső interf
- megadom a POOL-t benne ip cím és fordított maszk
- belső címek engedése - access-list készítépse
- összerendelem a POOL-t és a access-list-et
- CSAK ANNYICÍM LEHET, AMENNYI A CÍMKÉSZLETBEN VAN//overload

```
CORPORATE(config)#int gig0/2
CORPORATE(config-if)#ip nat inside
CORPORATE(config-if)#exit
CORPORATE(config)#interf ser0/0/0
CORPORATE(config-if)#ip nat outside
CORPORATE(config-if)#exit

CORPORATE(config)#access-list 9 permit 10.0.0.0 0.255.255.255
CORPORATE(config)#ip nat pool callcenter 188.18.100.3
188.18.100.12 netmask 255.255.255.240
CORPORATE(config)#ip nat inside source list 9 pool callcenter
```

Alapértelmezett útvonal megadása a külvilág eléréséhez:
R1(config)#ip route 0.0.0.0 0.0.0.0 ser 0/0/0

NAT ellenőrzése

A statikus NAT ellenőrzése

```
show ip nat translations
show ip nat statistics
clear ip nat statistics
```

A dinamikus NAT ellenőrzése

```
show ip nat translations (verbose)
ip nat translation timeout elévülési-idő-másodpercben
show ip nat statistics
clear ip nat statistics
```

A PAT ellenőrzése

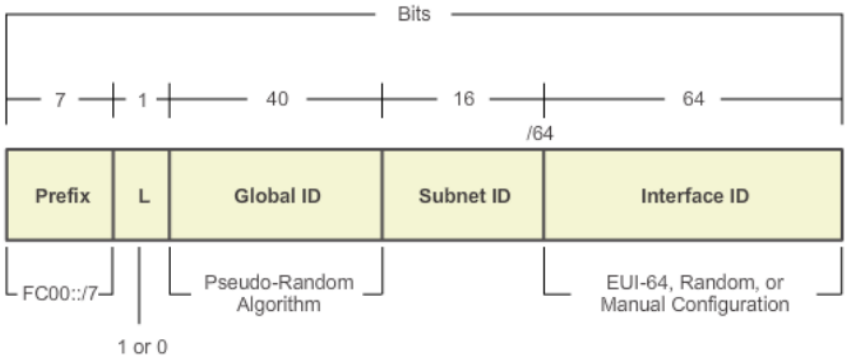
```
show ip nat translations (verbose)
show ip nat statistics
```

És mindenhhol a
show running-config

NAT IPV6 esetén

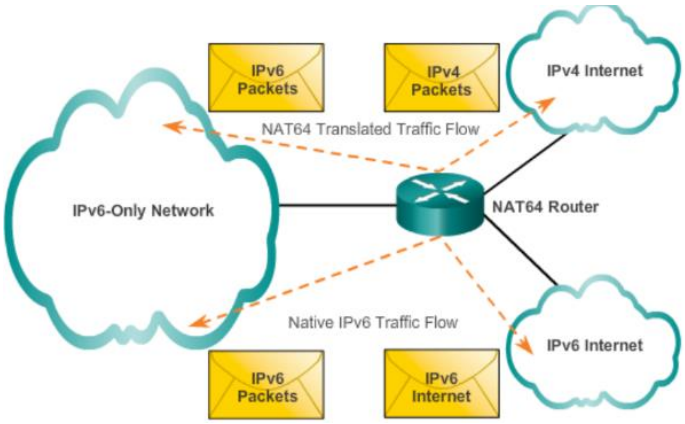
- A 128 bites IPv6 címekből 340 undecillion létezik.
- A címtartomány mérete nem jelenthet problémát IPv6 esetén.
- Az IPv4-nél megismert publikus-privát címek közötti NAT szükségtelen az IPv6 struktúrája miatt (emiatt is
- tervezték ilyenre); ennek ellenére az IPv6-ban is van privát címtartomány, csak másként valósították meg.
- (however, IPv6 does implement a form of private addresses, and it is implemented differently than they are
- for IPv4.)

- Az IPv6 Unique Local Address (ULA) típusú címeket tervezték egy szervezet belső hálózatai közötti kommunikációra.
- Az ULA címek nem jelentenek bővítést az IPv6 címtérben
- Az ULA címek az FC00::/7 tartomány alá esnek, így az ilyen címek első hextettje FC00 – FDFF lehet.
- Az ULA-t másként helyi IPv6 címekként ismerjük (nem keverendő össze a link-local címekkel).



Az IPv6-ban is ismerünk NAT-ot, csak más értelemben

- Az IPv6-ban a NAT az átjárhatóságot teremti meg az IPv6 és az IPv4 között: ezért hívják NAT64-nek
- A NAT64 nem végleges megoldásként készült, inkább egyfajta átállást segítő mechanizmus.
- Többféle NAT-ot alkottak az IPv6-tal kapcsolatban, ilyen pl. a „Hálózati címfordítás – protokoll fordítás”
- (Network Address Translation – Protocol Translation; NAT-PT), de ezt az IETF elavultnak jelölte.

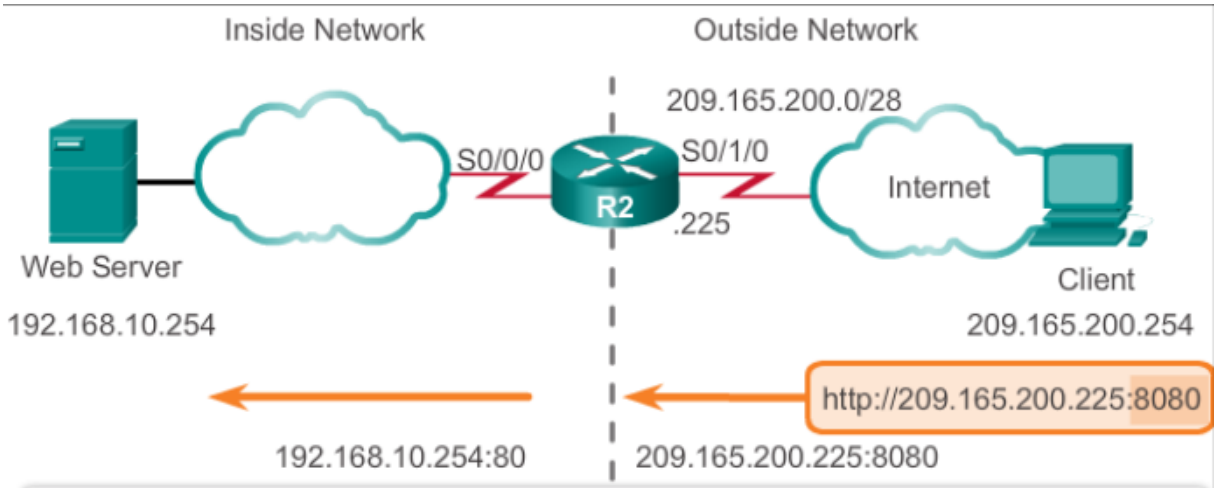


- Jelenleg a NAT64 a javasolt, ha valamilyen megoldás kell a kétféle címzési tér között.

Port továbbítás (port-forwarding)

- Az egyik hálózaton megcímzett port továbbítása a másik hálózaton egy adott eszköz adott portjára
- Tehát a csomagot a router publikus címére egy adott portra címzik, ezt a router átfordítja a belső hálózatra egy belső eszköz belső címére és adott portjára
- Akkor nagy segítség, ha egy szerver csak belső címmel rendelkezik, és egy-egy portján szolgáltat csak

Az IOS-ben a Port-továbbítás lényegében statikus NAT-fordítás, meghatározott TCP- vagy UDP-portszámmal.



Establishes static translation between an inside local address and local port and an inside global address and global port.

```
R2(config)# ip nat inside source static tcp 192.168.10.254 80
209.165.200.225 8080
```

Identifies interface serial 0/0/0 as an inside NAT interface.

```
R2(config)# interface Serial0/0/0
R2(config-if)# ip nat inside
```

Identifies interface serial 0/1/0 as the outside NAT interface.

```
R2(config)# interface Serial0/1/0
R2(config-if)# ip nat outside
```

Router(config)#ip nat inside source static beső_cím és port külső_cím és port

Hálózati redundancia

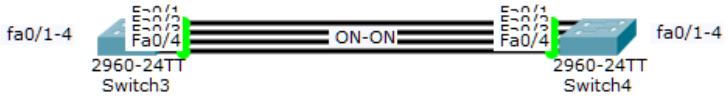
STP (Spanning Tree Protocol)

Hídprioritás beállítása (az érték 0-61440 között lehet, 4096-os lépésekkel, a kisebb lesz a gyökérponti híd):
Switch(config)#spanning-tree vlan 1 priority 4096
illetve:
Switch(config)#spanning-tree vlan 1 root [primary **|** secondary **]**
Hozzáférési portok gyors-továbbító üzemmódba állítása:
Switch(config)#spanning-tree portfast default
illetve interfészenként:
Switch(config-if)#spanning-tree portfast
A kialakult állapot megjelenítése:
Switch# show spanning-tree [detail | summary **|** vlan **x]**
Üzemmód beállítása (normál / gyors)
Switch(config)#spanning-tree mode pvst | rapid-pvst
Interface költség beállítás:
Switch(config-if)#spanning-tree vlan 10 cost 30
Alapértelmezett értékek: 10Mbps=100; 100Mbps=19; 1Gbps=4; 10Gbps=2
Root guard (hogy a gyökérponti kapcsoló ne változzon a hálózaton):
Switch(config)#spanning-tree guard root
Loop guard engedélyezése globálisan:
Switch(config)#spanning-tree loopguard default
BPDU guard engedélyezése globálisan (hogy bármilyen portról ne fogadjon bpdu-t):
Switch(config)#spanning-tree portfast bpduguard default
illetve adott hozzáférési porton:
Switch(config-if)#spanning-tree bpduguard enable
EtherChannel guard (EtherChannel hibák ellenőrzésére):
Switch(config)#spanning-tree etherchannel guard misconfig

EtherChannel beállítása

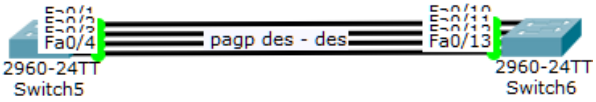
Manuális EC

```
Switch(config-if-range)#channel-group 1 mode on
```



PagP EC

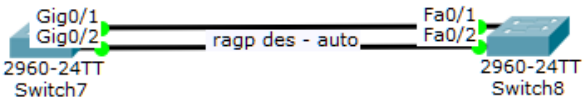
```
Switch(config-if-range)#channel-group 3 mode desirable
```



így alapból nem kapcsol be, csak ha mindkét oldadl készen van

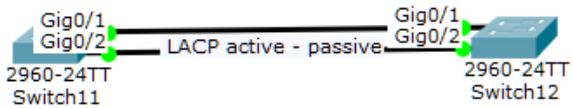
RagP EC

```
Switch1(config-if-range)#channel-group 4 mode desirable
Switch2(config-if-range)#channel-group 4 mode auto
```



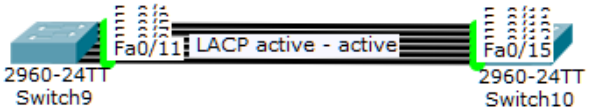
LACP EC

```
Switch1(config-if-range)#channel-group 6 mode active
Switch2(config-if-range)#channel-group 6 mode passive
```



Vagy

```
Switch(config-if-range)#channel-group 2 mode active
```



Használata interfészkén

```
interface port-channel 1
```

Ellenőrzés

```
show interface port-channel
show interface port-channel 5 brief
show interface port-channel 50 flowcontrol
show interface port-channel 50 switchport
show interface port-channel 5 trunk
show interface port-channel 2 counters
show interface port-channel 20 counters detailed all
show interface port-channel 5 counters errors
show interface port-channel 5 counters trunk
show interface status up
```

Pont-pont kapcsolat

Soros interface

```
(config)#interface Serial 0
(config-if)#ip address 193.155.145.2 255.255.255.0
(config-if)#encapsulation hdlc | ppp
(config-if)#clock rate 64000
(config-if)#no shutdown
(config-if)#exit
```

show controllesr itt látom ki a DCE ill. DTE

PPP – PAP

Elég csak az egyik oldalon hitelesíteni (de a másik oldalon is be lehet állítani).

A hitelesítés csak a csatlakozáskor történik meg.

```
Router(config)#hostname R1
R1(config)#username masik_router password paptitok
R1(config)#interface Serial 0/1/0
R1(config-if)#ip address 188.15.70.1 255.255.255.0
R1(config-if)#encapsulation ppp (gyárilag hdlc)
R1(config-if)#ppp authentication pap
R1(config-if)#ppp pap sent-username R1 password paptitok
R1(config-if)#no shutdown
```

PPP- -CHAP

A hitelesítést mindkét routeren be kell állítani. Majd a kapcsolat során többször is hitelesítik egymást.

```
(config)#hostname egyik_router
(config)#username masik_router password chaptitok
a jelszavaknak egyezni kell a routereken
(config)#interface Serial 1
(config-if)#ip address 188.15.70.1 255.255.255.0
(config-if)#encapsulation ppp
(config-if)#ppp authentication chap
(config-if)#no shutdown
```

Debug parancsok

```
Router# debug ppp authentication
Router#debug ppp packet
Router# debug ppp error
Router# debug ppp chap
```

Frame Relay – Virtuális körök

```
R1(config)# interface serial 0/0/1
R1(config-if)# ip address 10.1.1.1 255.255.255.0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no frame-relay inverse-arp
R1(config-if)# frame-relay map ip 10.1.1.2 102 broadcast
cisco
R1(config-if)# no shutdown
R1(config-if)#
*Mar 31 18:57:38.994: %LINK-3-UPDOWN: Interface Serial0/0/1,
changed state to up
R1(config-if)#
```

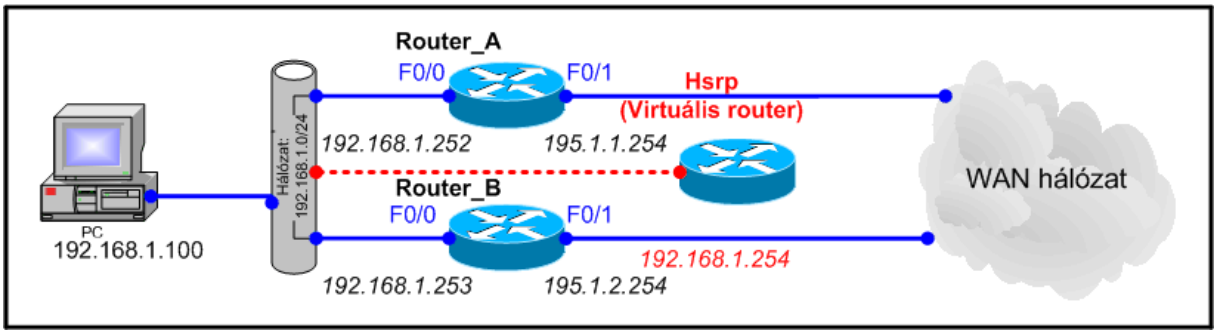
```
R1# show frame-relay map
Serial0/0/1 (up): ip 10.1.1.2 dlci 102(0x66,0x1860), static,
              broadcast,
              CISCO, status defined, active
R1#
```

```
R1# show frame-relay lmi

LMI Statistics for interface Serial0/0/1 (Frame Relay DTE)
LMI TYPE = CISCO
  Invalid Unnumbered info 0Invalid Prot Disc 0
  Invalid dummy Call Ref 0Invalid Msg Type 0
  Invalid Status Message 0Invalid Lock Shift 0
  Invalid Information ID 0Invalid Report IE Len 0
  Invalid Report Request 0Invalid Keep IE Len 0
  Num Status Enq. Sent 368Num Status msgs Rcvd 369
  Num Update Status Rcvd 0Num Status Timeouts 0
  Last Full Status Req 00:00:29Last Full Status Rcvd 00:00:29
R1#
```

Virtuális Router beállítása

HSRP - Hot Standby Router Protocol

Feladata hibatűrő routolt hálózat létrehozása több router felhasználásával. Egynél több router szükséges hozzá. Az egyik router kiesése esetén a tartalék router veszi át a forgalomirányítás szerepét. HSRP konfigurálásakor a fizikai routerek konfigurációja felett egy virtuális routert kell létrehozni.



C2600#Standby 12 ip 192.168.1.254

Engedélyezi a HSRP-t. A megadott IP cím a virtuális router címe, amelyet a HSRP csoport minden tagjában definiálni kell. Opcionálisan meg lehet adni a HSRP csoport számát is, amely alapértelmezett értéke 0 (nulla).

C2600#standby 12 preempt

Megengedi, hogy az adott router aktív legyen, abban az esetben ha a prioritása nagyobb, mint a HSRP csoporton belül a többi routeré. Ha a parancsot nem adjuk ki, akkor a router nem lehet aktív a HSRP csoportban. Opcionálisan a csoport száma is megadható

C2600#standby 12 priority 130

Az adott router prioritását adja meg. Az alapértelmezett szint 100. A csoporton belül a legnagyobb prioritású lesz az aktív router. Ha routernek nagyobb prioritást adunk, akkor nagyobb eséllyel lesz az adott router aktív. Opcionálisan a HSRP csoport száma is megadható.

C2600#standby authentication szoveg

Konfigurációs parancs, amelyben egy 8 karakteres szöveg megadható. Ez a szöveg belekerül a HSRP multicast csomagokba. Nem kötelező alkalmazni. Amennyiben használjuk, akkor a csoport minden tagjában alkalmazni kell, mert a csoport tagjai ezen szöveg alapján hitelesítik a csomagot. Opcionálisan a HSRP csoport száma is megadható.

C2600#standby 1 timers 6 18

konfigurációs parancs a HSRP belső időzítések idejének módosítására. Az értékek másodpercben értendőek. A router rendszeres időközönként (hello time) életjelet küld a hálózaton, amely időközöket az első paraméterben adhatunk meg. Alapértelmezett értéke 3 másodperc. A második paraméterben adható meg, hogy a router meddig várjon az életjelre.

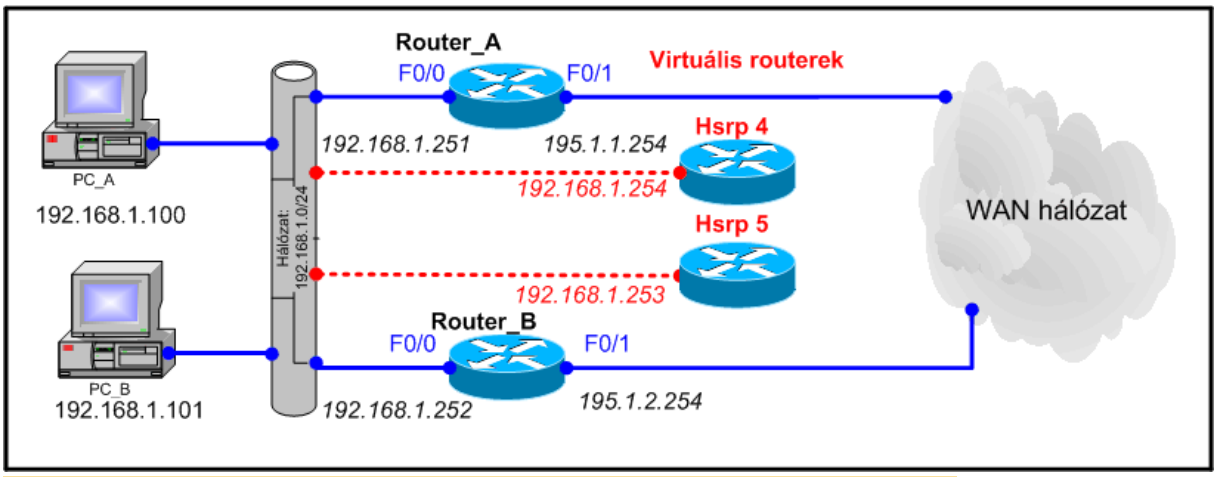
Alapértelmezett értéke 10 másodperc. A várakozási idő leteltével amennyiben a router nem kapott életjelet, a router az aktív routert nem működőnek tekinti. A standby timers parancsot ha megadjuk, akkor a HSRP csoporton belül a hello time és a hold time paramétereket egységesen kell beállítani. Opcionálisan a HSRP csoport száma is megadható.

```
hostname Router_A
...
interface FastEthernet 0/0
ip address 192.168.1.252 255.255.255.0
standby ip 192.168.1.254               ! HSRP virtuális router cím
standby preempt                        ! Aktív routerként engedélyezve
standby priority 110                   ! Default aktív routerként kiválasztva
!
interface FastEthernet 0/1
ip address 195.1.1.254 255.255.255.0
...
router eigrp 44
network 192.168.1.0
network 195.1.1.0
...
hostname Router_B
...
interface FastEthernet 0/0
ip address 192.168.1.253 255.255.255.0
standby ip 192.168.1.254               ! HSRP virtuális router cím
standby preempt                        ! Aktív routerként engedélyezve
!
interface FastEthernet 0/1
ip address 195.1.2.254 255.255.255.0
...
router eigrp 44
network 192.168.1.0
network 195.1.2.0
...
```

```
IP-cím. . . . . . . . . . . . : 192.168.1.100
Alhálózati maszk. . . . . . . : 255.255.255.0
Alapértelmezett átjáró. . . : 192.168.1.254
```

A példában a Router_A az alapértelmezett router, mivel a prioritása 110, és ez nagyobb mint a Router_B esetében. A router_B prioritása nem dekralált, igy az alapértelmezett 100-as értéket kapja meg. A PC konfigurációja függetlenné válik a routerek állapotától. Az alapértelmezett átjáró a virtuális router címére mutat.

HSRP konfiguráció terhelésmegosztással




```
interface FastEthernet 0/0
ip address 192.168.1.251 255.255.255.0
standby 4 ip 192.168.1.254
standby 4 priority 110
standby 4 preempt
standby 5 ip 192.168.1.253
standby 5 preempt
…
```

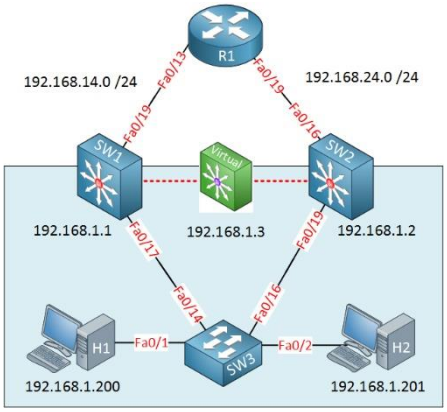
```
interface FastEthernet 0/0
ip address 192.168.1.252 255.255.255.0
standby 4 ip 192.168.1.254
standby 4 preempt
standby 5 ip 192.168.1.253
standby 5 priority 110
standby 5 preempt
…
```

```
IP-cím. . . . . . . . . . . . : 192.168.1.100
Alhálózati maszk. . . . . . . : 255.255.255.0
Alapértelmezett átjáró. . . : 192.168.1.254
```

```
IP-cím. . . . . . . . . . . . : 192.168.1.101
Alhálózati maszk. . . . . . . : 255.255.255.0
Alapértelmezett átjáró. . . : 192.168.1.253
```

A példában két HSRP csoport van. A 4-es csoport aktív tagja a Router_A és a meleg tartalék a Router_B. Az 5-ös csoportban viszont a Router_B aktív, a Router_A a melegtartalék. A Router_A kiesése esetén a 4-es csoportban a Router_B átveszi az aktív router szerepét, az 5-ös csoportban a szerepe nem változik, akív marad ott is. A Router_B kiesése esetén a 4-es csoportban a Router_A marad, és az 5-ös csoportban válik aktívvá. A hiba elmúlta esetén a routerek ismét az alapértelmezett szerepüket veszik vissza. A hibátlan működéshez a routerek konfigurálásakor minden csoportban ki kell adni a standby preempt parancsot. A PC-k terhelés megosztás alapu konfigurálásakor a gépek egyik felét az egyik viruális, a másik felét a másik virtuális routerhez kell konfigurálni.

VRRP - Virtual Router Redundancy Protocol




```
SW1(config)#interface fa0/17
SW1(config-if)#vrrp 1 ip 192.168.1.3
SW1(config-if)#vrrp 1 priority 150
SW1(config-if)#vrrp 1 authentication
md5 key-string mykey
SW2(config-if)#interface fa0/19
SW2(config-if)#vrrp 1 ip 192.168.1.3
SW2(config-if)#vrrp 1 authentication md5 key-string mykey
```

HSRP és VRRP összehasonlítása

	HSRP	VRRP
Protocol	Cisco proprietary	IETF – RFC 3768
Number of groups	16 groups maximum	255 groups maximum
Active/Standby	1 active, 1 standby and multiple candidates.	1 active and several backups.
Virtual IP Address	Different from real IP addresses on interfaces	Can be the same as the real IP address on an interface.
Multicast address	224.0.0.2	224.0.0.18
Tracking	Interfaces or Objects	Objects
Timers	Hello timer 3 seconds, hold time 10 seconds.	Hello timer 1 second, hold time 3 seconds.
Authentication	Supported	Not supported in RFC 3768

Ellenőrzés

```
Router# show { hsrp | vrrp | glbp }
Router# show { hsrp | vrrp | glbp } brief
Router# show { hsrp | vrrp | glbp } 1
Router# show { hsrp | vrrp | glbp } fastethernet 0/0
Router# show { hsrp | vrrp | glbp } fastethernet 0/0 1
```

Licenc csomagok telepítése

sh version

Csomag engedélyezése:

```
Router(config)#license boot module c2900 technology-package
securityk9
Router(config)#do write
Building configuration...
[OK]
Router(config)#do reload
```

Letiltása:

```
Router(config)#license boot module c2900 technology-package
securityk9 disable
```

VoIP telefonok használata

CME

CME

Telephony-service beállítása:

```
R1(config)#telephony-service
R1(config-telephony)#max-ephones 3 (telefonok száma)
R1(config-telephony)#max-dn 3 (telefonszámok száma)
R1(config-telephony)#ip source-address 10.1.1.1 port 2000
R1(config-telephony)#auto assign 1 to 3
R1(config-telephony)#create cnf-files version-stamp Jan 01
2002 00:00:00
R1(config-telephony)#max-conferences 4
R1(config-telephony)#transfer-system full-consult
```

Telefon beállítása egy illetve többvonalasra:

```
CME(config)#ephone-dn 5 ?
dual-line dual-line DN (2 calls per line/button) <cr>
```

Vonalak megadása:

```
R1(config)#ephone-dn 1 dual-line
R1(config-ephone-dn)#number 3000
```

Vonalak gombokhoz rendelése:

```
R1(config)#ephone 1
R1(config-ephone)#mac-address 0012.17F0.A883
R1(config-ephone)#type CIPC
R1(config-ephone)#button 1:5 3:6 4:7
```

Egy telefon újraregisztrálása

```
R0(config)#ephone 1
R0(config-ephone)#restart
```

A szükséges állományokat a Flash-be fel kell tölteni és be kell állítani az elérésüket

Az elérés beállítása

```
Router (config)# ip http server
Router (config)# ip http authentication local
Router (config)# ip http path flash:
Router (config)# username cmeadmin privilege 15 secret cisco
Router (config)# line con 0
Router (config-line)# logging sync
Router (config-line)# end
```

Az állományok feltöltése és kibontása

```
Router# archive tar /xtract tftp://10.10.10.2/cme.tar flash:
```

A felhasználói neveket a telefonszámokhoz rendelhetjük az ephone-dn bejegyzésekben

```
CME (config)# ephone-dn 20
CME (config-ephone-dn)# name Nagy Jozsef
CME (config-ephone-dn)# exit
```

Névsorba rendezés vezeték név alapján

```
CME(config-telephony)# directory last-name-first
```

Új elem felvétele a telefonkönyvbe

```
CME(config-telephony)# directory entry 1 1599 name Corporate
Fax
```

beállított értékek megjelenítése

```
R1#sh telephony-service directory-entry
```

Gyorshívás

```
speed-dial 1 5000 label "Jozsi" speed-dial 2 5001 label "Peti"
```

Hívás továbbítás CLI-ből

```
CME(config-ephone-dn)# call-forward busy 1599
CME(config-ephone-dn)# call-forward noan 1599 timeout 25
```

Ez a parancs megadja, hogy milyen hosszú telefonszámokra irányítható át a hívás. Amennyiben ez

a szám 0, akkor letiltja az átirányítást!

```
CME(config-ephone-dn)# call-forward max-length 0
```

mely telefonszámokra alkalmazhatjuk a H 450.3 átirányítást

```
call-forward pattern <pattern>
```

A hívás átengedés

```
CME(config)# telephony-service
CME(config-telephony)# transfer-system {full-blind|full-
consult|local-consult}
```

A hívás várakoztatás

```
CME(config)# ephone-dn 50
CME(config-ephone-dn)# number 3001
CME(config-ephone-dn)# name Maintenance
```

```
CME(config-ephone-dn)# park-slot
CME(config-ephone-dn)# exit
```

A hívás átvétel

```
CME(config)# ephone-dn 1
CME(config-ephone-dn)# pickup-group 5509
CME(config-ephone-dn)# ephone-dn 2
CME(config-ephone-dn)# pickup-group 5509
CME(config-ephone-dn)# ephone-dn 3
CME(config-ephone-dn)# pickup-group 5509
CME(config-ephone-dn)# ephone-dn 4
CME(config-ephone-dn)# pickup-group 5510
CME(config-ephone-dn)# ephone-dn 6
CME(config-ephone-dn)# pickup-group 5510
```

A tárcsázási párok beállítása

```
CME(Config)# dial-peer voice címke pots
```

Miután kialakítottunk egy tárcsázási párt, szükséges hozzárendelni a telefonszámot, és az egészet össze kell rendelnünk egy Voice porttal

```
CME(config-dial-peer)# destination-pattern 1102
CME(config-dial-peer)# port 2/0
```

A tárcsázás ellenőrzése

```
show dial-peer voice summary
```

Hívás nyomon követése

```
CME# debug voip dialpeer
```

A telefonszámok feldolgozása

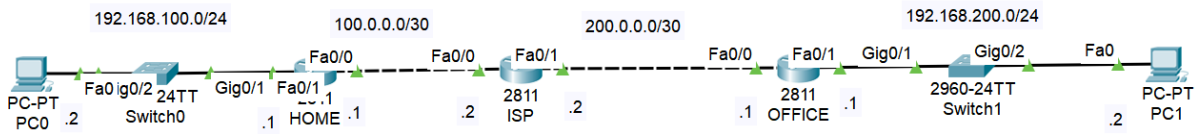
```
ROUTER_B(config-dial-peer)# destination-pattern 9
ROUTER_B(config-dial-peer)# no digit-strip
```

A tárcsázási párok beállítása

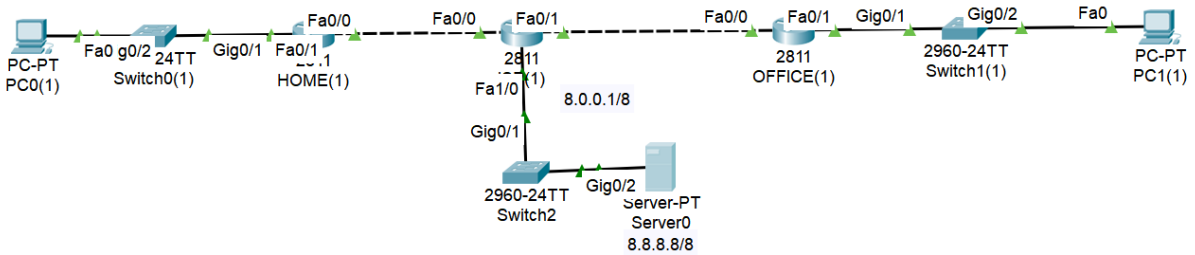
```
CME(config)# dial-peer voice 2000 voip
CME(config-dial-peer)# destination-pattern 2
CME(config-dial-peer)# session target ipv4:10.1.1.2
CME(config-dial-peer)# codec g711ulaw
ROUTER_B(config)# dial-peer voice 1100 voip
ROUTER_B(config-dial-peer)# destination-pattern 110
ROUTER_B(config-dial-peer)# session target ipv4:10.1.1.1
ROUTER_B(config-dial-peer)# codec g711ulaw
```

VIRTUÁLIS MAGÁN HÁLÓZATOK

GRE tunnel



interface FastEthernet0/0 ip address 100.0.0.2 255.255.255.252 interface FastEthernet0/1 ip address 200.0.0.2 255.255.255.252	
interface Tunnel1 ip address 192.168.0.1 255.255.255.0 tunnel source FastEthernet0/0 tunnel destination 200.0.0.1 ! interface FastEthernet0/0 ip address 100.0.0.1 255.255.255.252 ! interface FastEthernet0/1 ip address 192.168.100.1 255.255.255.0 ! ip route 192.168.0.0 255.255.255.0 100.0.0.2 ip route 200.0.0.0 255.255.255.252 100.0.0.2 ip route 192.168.200.0 255.255.255.0 192.168.0.2	interface Tunnel1 ip address 192.168.0.2 255.255.255.0 tunnel source FastEthernet0/0 tunnel destination 100.0.0.1 ! interface FastEthernet0/0 ip address 200.0.0.1 255.255.255.252 ! interface FastEthernet0/1 ip address 192.168.200.1 255.255.255.252 ! ip route 192.168.0.0 255.255.255.0 200.0.0.2 ip route 100.0.0.0 255.255.255.252 200.0.0.2 ip route 192.168.100.0 255.255.255.0 192.168.0.1



no ip route 192.168.0.0 255.255.255.0 100.0.0.2 no ip route 200.0.0.0 255.255.255.252 100.0.0.2	no ip route 192.168.0.0 255.255.255.0 200.0.0.2 no ip route 100.0.0.0 255.255.255.252 200.0.0.2
ip route 0.0.0.0 0.0.0.0 fastEthernet 0/0 access-list 1 permit any	ip route 0.0.0.0 0.0.0.0 fastEthernet 0/0 access-list 1 permit any

`ip nat inside source list 1`
`interface fastEthernet 0/0`
`interfa fa0/0`
`ip nat outside`
`interfa fa0/1`
`ip nat inside`
	`ip nat inside source list 1`
	`interface fastEthernet 0/0`
	`interfa fa0/0`
	`ip nat outside`
	`interfa fa0/1`
	`ip nat inside`

VPN PPTP protokoll használatával

Virtual Private Dialup Network engedélyezése

```
R1(config)#vpdn enable
```

Virtual Private Dialup Network létrehozása

```
R1(config)#vpdn-group 1
R1(config-vpdn)#accept-dialin
R1(config-vpdn-acc-in)#protocol pptp
R1(config-vpdn-acc-in)#virtual-template 1
```

Virtuális interfész valós interfészhez kötése

```
R1(config)#interface Virtual-Template1
R1(config-if)#ip unnumbered FastEthernet 0/0
R1(config-if)#peer default ip address pool PPTP-Pool
R1(config-if)#no keepalive
R1(config-if)#ppp encrypt mppe 128
R1(config-if)#ppp authentication ms-chap ms-chap-v2
```

Helyi hálózaton használható IP címek megadása

```
R1(config)#ip local pool PPTP-Pool 192.168.0.20 192.168.0.25
```

VPN felhasználó létrehozása

```
R1(config)#username user1 password cisco
```

VPN L2TP over IPSec használatával

Virtual Private Dialup Network engedélyezése

```
R1(config)#vpdn enable
```

Virtual Private Dialup Network létrehozása

```
R1(config)#vpdn-group 1
R1(config-vpdn)#no l2tp tunnel authentication
R1(config-vpdn)#accept-dialin
R1(config-vpdn-acc-in)#protocol l2tp
R1(config-vpdn-acc-in)#virtual-template 1
```

Virtuális interfész valós interfészhez kötése

```
R1(config)#interface Virtual-Template1
R1(config-if)#ip unnumbered FastEthernet0/0
R1(config-if)#peer default ip address pool L2TP-Pool
R1(config-if)#ppp authentication ms-chap-v2
Helyi hálózaton használható IP címek megadása
R1(config)#ip local pool L2TP-Pool 192.168.0.20 192.168.0.25
```

Hitelesítés beállítása

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption 3des
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 2
R1(config-isakmp)#lifetime 3600
IPSec előre megosztott kulcs megadása
R1(config)#crypto isakmp keepalive 3600
R1(config)#crypto isakmp key cisco address 0.0.0.0 0.0.0.0 no-
xauth
```

IPSec beállítás

```
R1(config)#crypto ipsec transform-set MySet esp-3des esp-sha-
hmac
R1(cfg-crypto-trans)#mode transport
R1(config)#crypto dynamic-map MyMap 10
R1(config-crypto-map)#set transform-set MySet
R1(config)#crypto map L2TP-Map 10 ipsec-isakmp dynamic MyMap
R1(config)#interface FastEthernet0/0
R1(config-if)#crypto map L2TP-Map
VPN felhasználó létrehozása
R1(config)#username user1 password cisco
```

Site-to-Site VPN IPSec

ISAKMP konfiguráció

```
R1(config)#crypto isakmp policy 6
```

Hitelesítés

```
R1(config-isakmp)#authentication pre-share
```

Diffie-Hellman csoport

```
R1(config-isakmp)#group 5
```

Kivonatoló algoritmus

```
R1(config-isakmp)#hash md5
```

Titkosítás

```
R1(config-isakmp)#encr 3des
```

Az SA élettartama

R1(config-isakmp)#lifetime 3600

Közös titkos kulcs és másik végpont megadása

R1(config)#crypto isakmp key Secret address 200.20.2.1

IPSec globális SA élettartamának konfigurálás

R1(config)#crypto ipsec security-association lifetime seconds 86400

Crypto ACL konfigurálása

R1(config)#access-list 100 permit ip 192.168.0.0 0.0.255.255 10.0.0.0 0.255.255.255

Transzform set beállítása

R1(config)#crypto ipsec transform-set SETNAME esp-3des esp-md5-hmac

Crypto map konfigurlása

R1(config)#crypto map MAPNAME PRIORITY ipsec-isakmp

Társ végpont

R1(config-crypto-map)#set peer 200.20.2.1

Transzform set megadása

R1(config-crypto-map)#set transform-set SETNAME

DH group hozzárendelése

R1(config-crypto-map)#set pfs group5

Crypto ACL hozzárendelése

R1(config-crypto-map)#match address 100

Crypto map hozzárendelése VPN végpont interfészhez

R1(config-if)#crypto map MAPNAME

IPSEC VPN (Packet Tracer-ben működő)

VPN felhasználó létrehozása

R1(config)#username vpnuser password cisco

Csoport hozzáadása

R1(config)#aaa new-model

```
R1(config)#aaa authentication login default local
R1(config)#aaa authorization network default local
```

Csoport hozzáadása Radius hitelesítés esetén

```
R1(config)#aaa authentication login default group radius local
R1(config)#aaa authorization network default group radius
local
R1(config)#radius-server host 172.16.1.1 auth-port 1645 key
cisco
```

Helyi hálózaton használható IP címek megadása (ezek lesznek kiosztva)

```
R1(config)#ip local pool VPN-Pool 192.168.0.20 192.168.0.25
```

Hitelesítés beállítása

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption 3des
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 2
R1(config-isakmp)#lifetime 3600
```

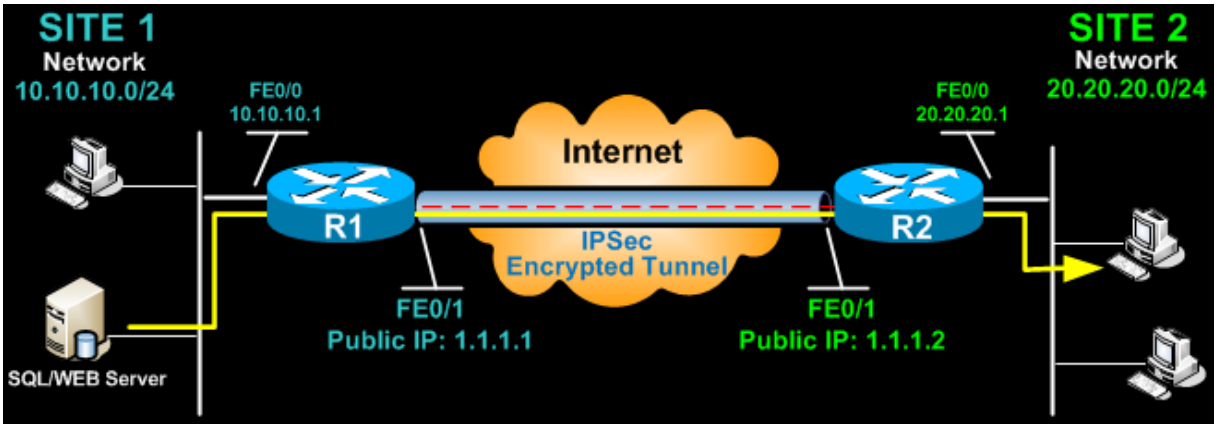
IPSec csoport létrehozása és konfigurálása

```
R1(config)#crypto isakmp client configuration group vpncsoport
R1(config-isakmp-group)#key cisco123
R1(config-isakmp-group)#netmask 255.255.255.0
R1(config-isakmp-group)#pool VPN-Pool
```

IPSec beállítás

```
R1(config)#crypto ipsec transform-set MySet esp-3des esp-sha-
hmac
R1(config)#crypto dynamic-map MyMap 10
R1(config-crypto-map)#set transform-set MySet
R1(config-crypto-map)#reverse-route
R1(config)#crypto map VPN-Map client authentication list
default
R1(config)#crypto map VPN-Map client configuration address
respond
R1(config)#crypto map VPN-Map isakmp authorization list
default
R1(config)#crypto map VPN-Map 10 ipsec-isakmp dynamic MyMap
R1(config)#interface FastEthernet0/0
R1(config-if)#crypto map VPN-Map
```

Site to Site VPN tunnel Beállítása



ISAKMP (IKE) - (ISAKMP PHASE 1)

1. ISAKMP PHASE 1 POLICY:

```
R1(config)#  crypto isakmp policy 1
R1(config-isakmp)# encr 3des
R1(config-isakmp)# hash md5
R1(config-isakmp)# authentication pre-share
R1(config-isakmp)# group 2
R1(config-isakmp)# lifetime 86400
R1(config)# crypto isakmp key firewallcx address 1.1.1.2
```

IPsec beállítása

Kiterjesztett ACL készítése

```
R1(config)# ip access-list extended VPN-TRAFFIC
R1(config-ext-nacl)# permit ip 10.10.10.0 0.0.0.255 20.20.20.0
0.0.0.255
```

IPsec TRANSFORM (ISAKMP PHASE 2 POLICY) készítése

```
R1(config)# crypto ipsec transform-set TS esp-3des esp-md5-
hmac
```

CRYPTO MAP készítése

```
R1(config)# crypto map CMAP 10 ipsec-isakmp
R1(config-crypto-map)# set peer 1.1.1.2
R1(config-crypto-map)# set transform-set TS
R1(config-crypto-map)# match address VPN-TRAFFIC
```

CRYPTO MAP alkalmazása a publikus interfészen

```
R1(config)# interface FastEthernet0/1
R1(config- if)# crypto map CMAP
```

R2-t is be kell állítani

```
R2(config)# crypto isakmp policy 1
R2(config-isakmp)# encr 3des
R2(config-isakmp)# hash md5
R2(config-isakmp)# authentication pre-share
R2(config-isakmp)# group 2
R2(config-isakmp)# lifetime 86400

R2(config)# crypto isakmp key firewallcx address 1.1.1.1
R2(config)# ip access-list extended VPN-TRAFFIC
R2(config-ext-nacl)# permit ip 20.20.20.0 0.0.0.255 10.10.10.0
0.0.0.255

R2(config)# crypto ipsec transform-set TS esp-3des esp-md5-
hmac
R2(config)# crypto map CMAP 10 ipsec-isakmp
R2(config-crypto-map)# set peer 1.1.1.1
R2(config-crypto-map)# set transform-set TS
R2(config-crypto-map)# match address VPN-TRAFFIC

R2(config)# interface FastEthernet0/1
R2(config- if)# crypto map CMAP
```

Címfordítás (NAT) és IPSEC VPN TUNNEL

```
R1(config)# ip nat inside source list 100 interface
fastethernet0/1 overload
R1(config)# access-list 100 remark -=[Define NAT Service]=-
R1(config)# access-list 100 deny ip 10.10.10.0 0.0.0.255
20.20.20.0 0.0.0.255
R1(config)# access-list 100 permit ip 10.10.10.0 0.0.0.255 any
R1(config)# access-list 100 remark

R2(config)# ip nat inside source list 100 interface
fastethernet0/1 overload
R2(config)# access-list 100 remark -=[Define NAT Service]=-
R2(config)# access-list 100 deny ip 20.20.20.0 0.0.0.255
10.10.10.0  0.0.0.255
R2(config)# access-list 100 permit ip 20.20.20.0 0.0.0.255 any
R2(config)# access-list 100 remark
```

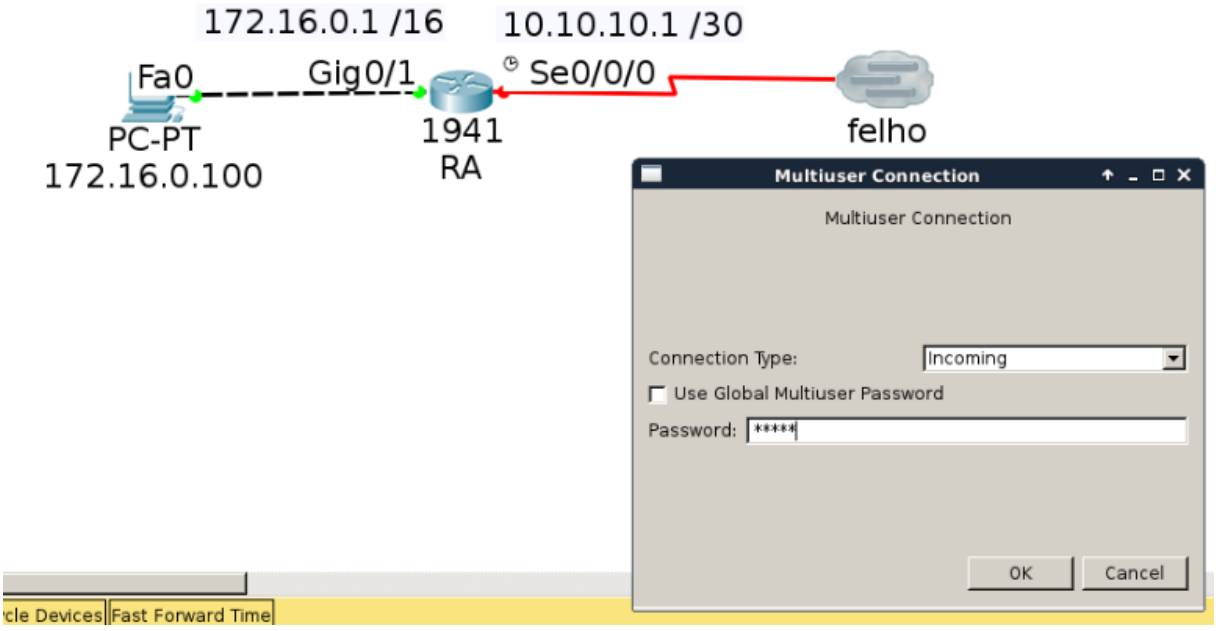
Ellenőrzés

```
R1# ping 20.20.20.1 source fastethernet0/0
R1# show crypto session
```

Multi User beállítása PT-ben

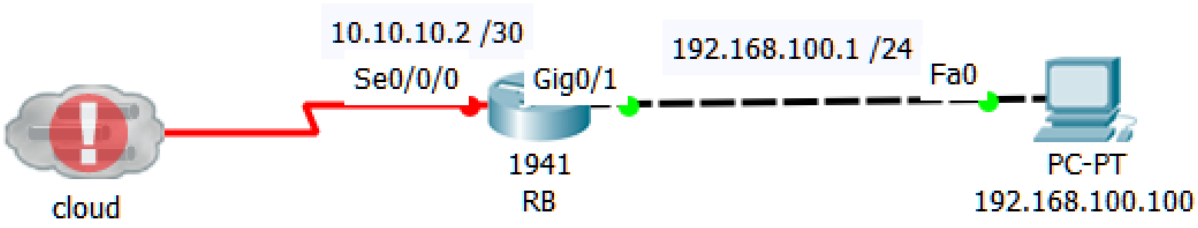
Szerver oldal

```
Router(config)#hostname RA
RA(config)#interf ser0/0/0
RA(config-if)#ip add 10.10.10.1 255.255.255.252
RA(config-if)#clock rate 64000
RA(config-if)#no shu
RA(config-if)#interf gig0/1
RA(config-if)#ip add 172.16.0.1 255.255.0.0
RA(config-if)#no shu

RA(config)#router OSPF 1
RA(config-router)#network 172.16.0.0 0.0.255.255 area 1
RA(config-router)#network 10.10.10.0 0.0.0.3 area 1
```

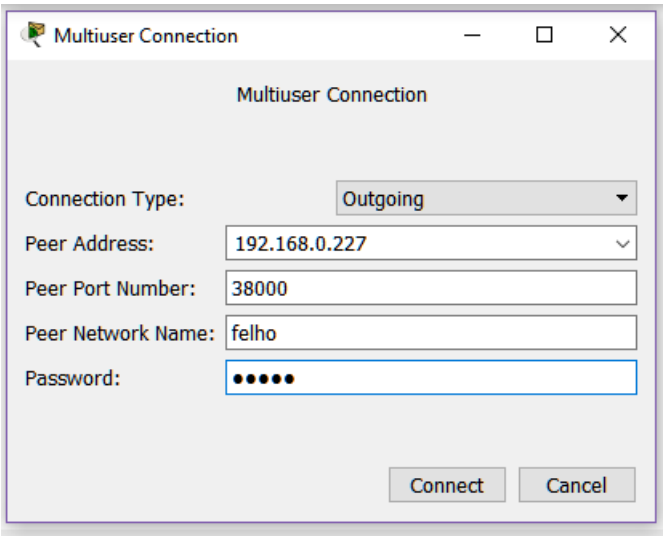


Kliens oldal

Hasonlóan de a végén



- A szerver gépen lévő felhő nevét kell megadni, mint felhasználó név
- A szervergép FIZIKAI IP címét kell használni
- A szervergépen megadott jelszót kell használni



Elmélet

OSI modell

A hálózatokra vonatkozó rétegmodell megfogalmazására 1980-ban került sor OSI (Open System Interconnection) néven.

Fontos tudni, hogy maga a modell nem szabvány, tehát nem egy ténylegesen megvalósítandó hálózat pontos leírása, csupán egy ajánlás, amely rögzíti, és rétegekbe rendezi a hálózati kommunikáció során megvalósítandó feladatokat.

Az OSI modell egyfajta „kályha" amelytől elindulva ki lehet dolgozni az egyes hálózati megvalósítások pontos rendszerét. Betartása nem kötelező. A megvalósított rendszerekben egyes rétegei szinte teljesen üresek maradnak, míg másoknál további osztásokra lett szükség zsúfoltságuk miatt. Hiányosságai ellenére a mai napig alapnak tekintik a gyártók.

Az OSI referencia modell szerint egy hálózatot 7 rétegre osztunk.

Az adatátvitellel foglalkozó rétegek:

1. A **fizikai réteg** (physical layer) A fizikai réteg a legalsó réteg, ezen zajlik a tényleges adatátvitel. Feladata a bitek hibamentes átvitele, azaz biztosítja, hogy az adó által küldött jeleket a vevő is azonosként értelmezze.

2. Az **adatkapcsolati réte**g (data link layer) Az adatkapcsolati réteg feladata az adatok kisebb egységekre, úgynevezett adatkeretekre (data frame) darabolása, és a keretek hibamentes célbajuttatása. Ezt úgy éri el, hogy a csomagokban adathalmazát egységnyi darabokra vágja, és majd minden kereten elvégez egy bonyolult matematikai műveletet, amelynek eredményét a keret végéhez illeszti. Ezt a számot CRC-nek (ciklikus redundancia control) nevezzük. A fogadó gép, miután megkapott egy keretet, ugyanazt a matematikai műveletet végzi el vele, mint a feladó gép. Saját eredményét összehasonlítja a keret végén található CRC-vel. Ha az elküldött, illetve a vevő oldalon számított eredmény megegyezik, akkor a vevő gép adatkapcsolati rétege egy úgynevezett nyugtakeretet küld a küldő gép adatkapcsolati rétegének, jelezve, hogy a keret hibamentesen megérkezett. Ha a küldő gép bizonyos időn belül nem kap nyugtakeretet, akkor az adatkeretet elveszettnek minősíti, és ismételten elküldi azt, forgalomszabályozást is végezve. A hibátlanul megérkező adatkereteket az adatkapcsolati réteg csomaggá illeszti össze, majd továbbítja azt a hálózati rétegnek.

3. A **hálózati réteg** (network layer) Vezérli a kommunikációs alhálózatok működését, legfontosabb feladata az útvonalválasztás a forrás és célállomás között. Ha az útvonalban eltérő hálózatok is vannak, akkor protokollátalakítást, -tördelést (fragmentation) is végez. Fontos megjegyezni, hogy míg az adatkapcsolati réteg az egymással kommunikáló távoli gépek között tartja a kapcsolatot és nem vesz tudomást az „útközben" elhelyezkedő gépekről, addig a hálózati réteg mindig csak egy szomszédos hosttal van kapcsolatban.

4. A **szállítási réteg** (transport layer) A végpontok közötti hibamentes adatátvitel biztosításáért felelős. A topológiát már nem ismeri, csak a két végpontban van rá szükség.

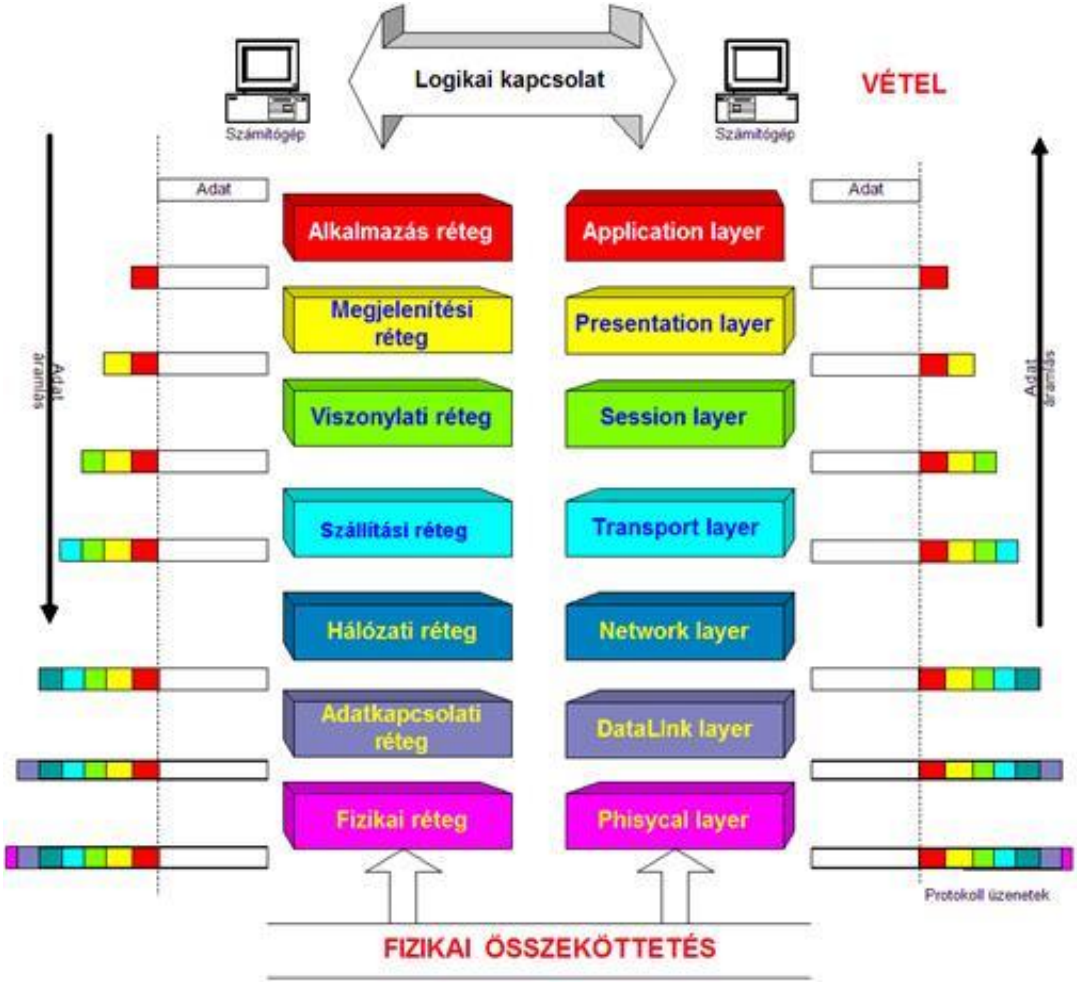
Feladatai: összeköttetések felépítése, bontása, csomagok sorrendbe állítása, hibaérzékelés, helyreállítás és az adatáramlás vezérlése.

A logikai összeköttetéssel foglalkozó rétegek:

5. A **viszonyréteg** (session layer) Megteremti annak a lehetőségét, hogy két számítógép felhasználói kapcsolatot létesítsenek egymással, azaz a programok, pontosabban folyamatok összekapcsolását végzi el. Feladata az alkalmazások közti viszonyok felépítése, kezelése és lebontása.

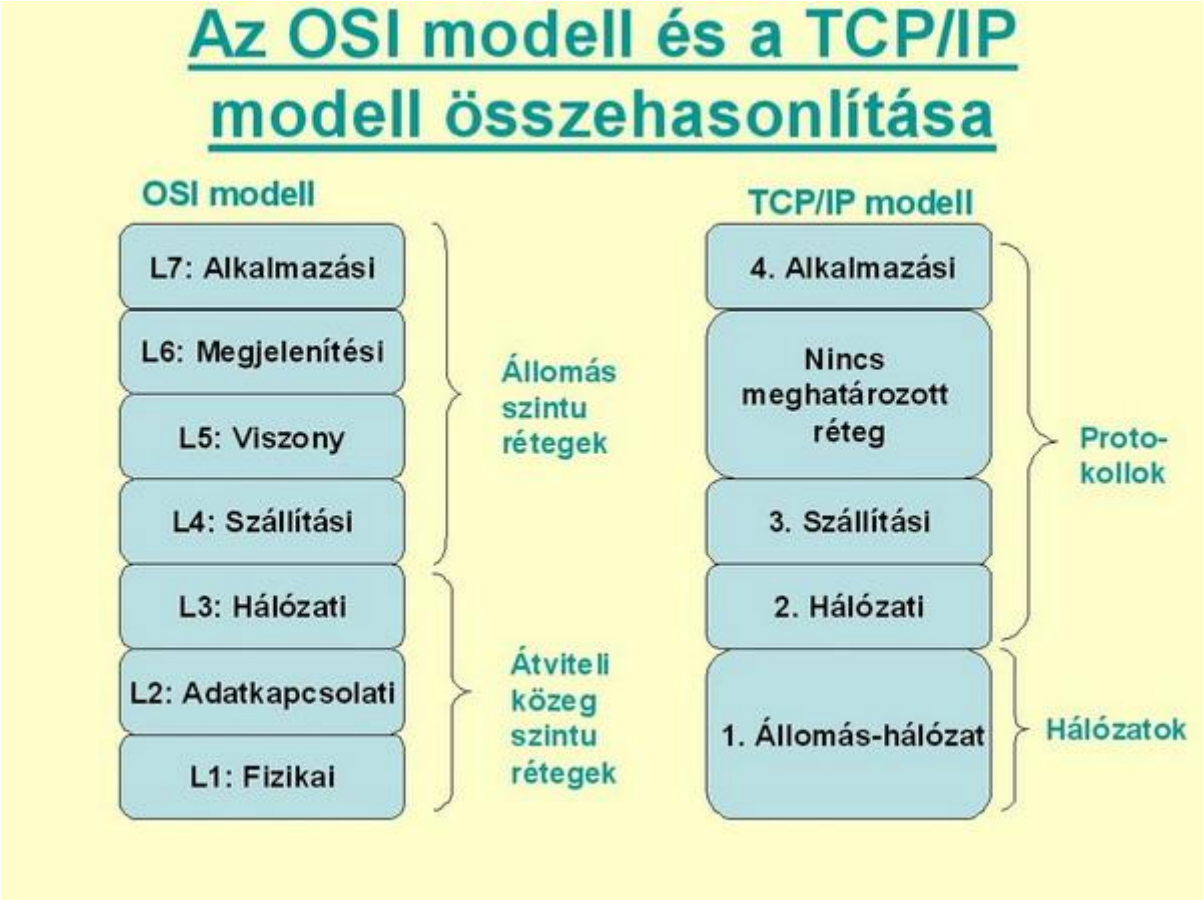
6. A **megjelenítési réteg** (presentation layer) A fogadó rendszer számára biztosítja az adatok olvashatóságát. A megjelenítési réteg feladatai közé tartozik az adatok titkosítása, és visszafejtése is. A rétegek közül az egyetlen, amely megváltoztathatja az üzenet tartalmát.

7. Az **alkalmazási réteg** (application layer) Az alkalmazások számára biztosít hálózati szolgáltatásokat. Az adó oldalon elfogadja és feldolgozza a felhasználó által továbbítandó adatokat, a vevő oldalon pedig gondoskodik azok felhasználó felé történő továbbításáról. Pl.: fájlok gépek közötti másolása.



TCP/IP protokoll

Ez valójában nem egy, hanem két, egymással összekapcsolt protokoll rendszere: a TCP (Transmission Control Protocol - átviteli vezérlő eljárás) és az IP (Internet Protocol - internet eljárás) protokolloké. Csomagkapcsolt hálózatok adatátviteli eljárásaként hozták létre. Felépítésében követi az OSI modellt, de egyes rétegeket összevontan kezel.



Host és network réteg (az OSI modell fizikai és adatkapcsolati rétegének egyben történő megvalósításával)

Feladata az adatfolyam kezelése, a keretek hálózati forgalmának lebonyolítása. A fölötte lévő szintek ezeket az adatkereteket adják át adó, és fogadják vevő oldalon.

Internetwork réteg (az OSI modell hálózati rétegének felel meg)

Feladata az adatok átvitele a hálózaton, függetlenül annak útvonalától. Gyakran hasonlítják ezt a postai szolgáltatáshoz: amikor egy levelet feladunk, a borítékra csak a feladó és a címzett adatai kerülnek rá, és nem tudjuk - de nem is fontos számunkra -, hogy az miként, milyen úton érkezik meg a rendeltetési helyére. Ezen a szinten három protokoll található:

- **IP (Internet Protocol)**

Az IP gondoskodik a hálózaton a csomagok átviteléről a hostok között. Ez egy kapcsolat nélküli protokoll, azaz a csomagok forgalmához nem szükséges a küldést megelőző

kapcsolatfelvétel. E miatt adatátviteli szempontból nem megbízható, hisz semmi nem garantálja, hogy a csomagok nem vesznek el, sorrendjük nem keveredik össze. Az IP csomag két fő részből áll:

IP fejléc

Adatmező

- **ARP (Adres Resolution Protocol)**

Az adatcsomag fizikai címének megkeresésére szolgál. Egy olyan IP csomagot hoz létre és továbbít adatszórásos (broadcast) elven, melyben egyaránt szerepel a keresett IP cím, a saját IP cím és a fizikai cím. Ha az IP cím alapján egy eszköz magát azonosítja, saját fizikai címével a csomagot kiegészíti, s a csomagot visszaküldi az eredeti feladónak.

- **ICMP (Internet Control Message Protocol)**

Hibajelzésre és a kapcsolatban álló két fél egyéb paramétereinek elküldésére szolgál. Ez is IP csomagként halad a hálózaton.

Transport réteg (az OSI modell szállítási rétegével azonosítható)

Az alkalmazási rétegtől kapott adatot a küldéshez szükséges fejjel egészíti ki. Kétféle, egymástól teljesen független protokollt használ:

- **TCP (Transmission Control Protocol)**

Ez az átvitel-vezérlési eljárás. Kezdeményezőként (adóként) küld egy kérés csomagot, bevárja a címzett válaszát, s ezt az adó egy nyugtázó üzenettel hálálja meg. A kapcsolatok azonosítására a portok szolgálnak, az első 1024 TCP port foglalt a standard alkalmazások számára. Figyeli a csomagok sorrendjét is. Ha a csomagok sorrendjétől eltérően egy későbbi csomagról kap nyugtát (egy vagy több csomag nyugtázása kimarad), akkor a sorfolytonosan legutolsó nyugtázott csomagot követő csomagtól kezdve megismétli az adást.

- **UDP (User Datagram Protocol)**

Mivel ez az eljárás nem kapcsolathoz kötött, így nincs nyugtázás és hibajavítás sem. Cserébe sokkal gyorsabb adatátvitelt tesz lehetővé. Az olyan alkalmazások használják, amelyeknél a pontosságnál fontosabb a gyorsaság. (Pl. valós idejű hangátvitel esetén kisebb gond, ha pillanatnyi hanghiba adódik, mintha a folyamat egésze időcsúszást szenved.)

Alkalmazási réteg (az OSI modell viszony-, megjelenítési és alkalmazási rétegeinek egy szinten való megvalósítása)

Ezen a szinten találjuk az alkalmazásokat. Az adatfolyamot a szállítási rétegnek továbbítják, illetve attól fogadják. Ezt a TCP vagy UDP meghatározott portján keresztül valósítják meg. A standard portokhoz vannak olyan szabványosított eljárások rendelve, mint például: SMTP, POP3, FTP, http stb.

Az internet címzési rendszere: IP

A TCP/IP protokollstruktúrából következik, hogy minden, a hálózathoz csatlakozó számítógépnek rendelkeznie kell egy egyedi azonosítóval. Ez az IP cím. E szakaszban az IPv4 példáján mutatjuk be az IP címek rendszerét, azonban megemlítjük, hogy a hálózati eszközök egyre nagyobb száma szükségessé tette egy új azonosítási rendszer bevezetését (IPv6). Jelenleg a két szisztéma egymás mellett él.

Az IP cím 4 bájton (azaz 32 biten) tárolt információ, szemléletesen az egyes bájtok értékét pontokkal elválasztva írjuk le (pl. 192.168.190.2).

A címek felépítése hierarchikus rendet képez. Két részre bontjuk:

Hálózati azonosító

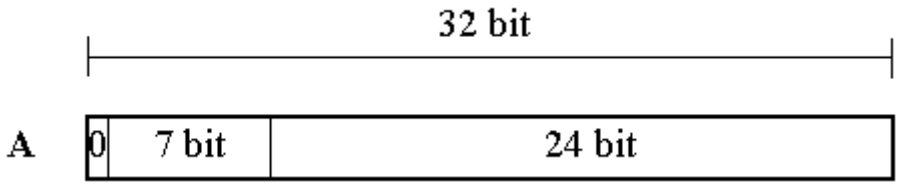
Ez egy hálózatot jelöl ki, a cím első (változó hosszúságú) része azonosítja. Az ütközések elkerülése érdekében ezt a NIC (Network Information Center) adja ki, s tartja nyilván.

Host azonosító

A hálózati azonosító által meghatározott hálózatban működő célállomás.

Címosztályok:

1. **A osztályú címek**



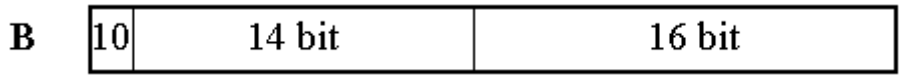
Az 1.0.0.0 és a 127.0.0.0 közötti hálózatokat foglalja magába. A hálózatot az első bájt, azaz a cím első számjegye azonosítja. A további három bájton ábrázolt értékeket a hálózatot felügyelő szabadon osztja ki. Olyan hálózatoknál alkalmazzák, amelyekben 65 536-nál több gép található. Ezzel a címzéssel mintegy 1,6 millió célállomás címezhető.

Példa A osztályú IP állomás címére: 120.10.20.30

10.0.0.0 – belső hálózatokban lehet használni (Intranet);

127.0.0.0 – belső hálózati tesztelési címek (loopback).

2. **B osztályú címek**

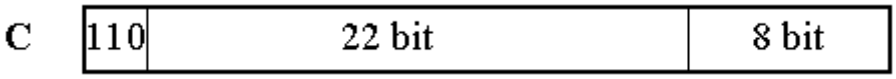


A 128.0.0.0 és a 191.255.0.0 közötti hálózatokat foglalja magába. A hálózatot az első 16 bit, azaz a cím első két számjegye azonosítja. Az állomások címei a másik 16 biten kiosztott címeken osztozhatnak, ezért ilyet csak a 65 536-nál kevesebb állomást tartalmazó hálózatok esetében lehet alkalmazni.

Példa B osztályú IP állomás címére: 168.10.20.30

172.16.0.0 – 172.31.0.0 – belső hálózatokban lehet használni (Intranet).

3. **C osztályú címek**



A 192.0.0.0 és 223.255.255.0 közötti hálózatokat foglalja magába. A célállomások címei mindössze 1 bájtnyi címterületen oszthatóak, így az ilyen hálózatokban legfeljebb 256 állomás címezhető.

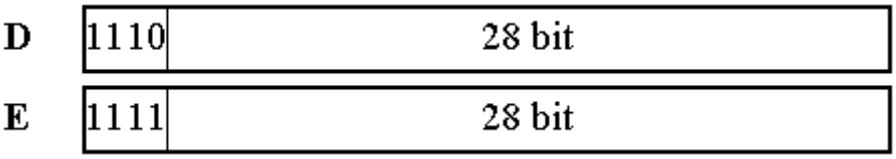
Példa C osztályú IP állomás címére: 120.10.20.30

192.168.1.0 – 192.168.255.0 – belső hálózatokban lehet használni (Intranet).

4. **Speciális IP címek**

A 224.0.0.0 és 254.0.0.0 a D, E és F kategóriába sorolt tartományi címeket nem adják ki hálózatok számára, azok speciális felhasználásra vannak fenntartva.

A 127.0.0.0 hálózat a helyi hálózati elemek tesztelési céljára szintén fenn van tartva. A működőképesség ellenőrzésére önmaguknak tudnak csomagot küldeni az eszközök a 127.0.01 címre.



„D” osztály - a 224.0.0.0 - 239.0.0.0 közötti címek tartoznak hozzájuk, multicasting eljárás céljaira vannak fenntartva.

„E” osztály - a 240.0.0.0 - 255.0.0.0 közötti címek tartoznak hozzájuk, melyek az Internet saját céljaira fenntartott címek.

Broadcast (szórási) címek: az adott hálózat IP címtartományának utolsó címét az állomások szórási címként használhatják. Azaz ha erre a címre csomagot küldenek (pl. útvonal-választási információ céljából említettük ezt az esetet a TCP/IP protokollstruktúra bemutatásakor), akkor azt minden, a hálózathoz csatlakozó állomás egyszerre megkapja.

Bár az előbb vázolt címzési rendszerrel 232 számú (mintegy 4 294 967 296) állomás címezhető (elvileg), e szám, bármilyen nagynak is tűnik, fogyóban van. Hisz állomáscímeket nem csak a számítógépek, hanem valamennyi, a hálózatot használó eszköz igényel. E probléma kiküszöbölésére is alkalmas megoldás az alhálózatok létrehozása.

Az elválasztást az alhálózati maszk szolgálja. Ez szintén egy 32 bites cím, ahol a hálózatnak megfelelő és az alhálózatot azonosító biteken 1-esek, a többi helyen 0-ák állnak (pl. C osztályú IP cím esetén 255.255.255.0). Ahhoz, hogy eldönthessük, egy számítógép mely hálózathoz tartozik, egy egyszerű ÉS (AND) műveletet kell végrehajtani az IP cím és a hálózati maszk értékeivel.

Az internet címzési rendszere

tartománynév

Az előzőekben bemutatott pontozott decimálisnak is mondott IP címek rendszere aligha sarkallna tömegeket hálózati kommunikációra, hisz az egyes címeket nehéz fejben tartani, nem lehet kötni a keresett információhoz. Ezért az egyes IP címekhez domain neveket (tartományneveket) rendelnek. Ezek nem maguk az adott weboldalak, hanem csupán egy felhasználóbarát címzési rendszer.

A tartománynevek felépítése (jobbról balra haladva):

Felső szintű tartomány (TLD - Top Level Domain)

Ez a név végén ponttal elválasztott utolsó elem. Két csoportja van:

o nemzetközi fődomain (top level domain): rendszerint három karakteres, a működési területet jelző rövidítés; ilyen például: .com, .org, .mil, de a .eu is ide sorolandó.

o nemzeti domainek: az adott ország nevére utaló két karakteres rövidítés; így a .hu Magyarországot, a . en és a .gb Angliát jelenti, és sorolhatnánk...

A második szintű tartomány (SLD)

A tartománynév e részlete szabadon megválasztható. Bár lehetőség van újabban nemzeti karakterkészleteket tartalmazó tartománynevek választására, ez - szerencsére - nem terjed, gondoljunk azokra a nehézségekre, melyek egy-egy magyar ékezetes betű begépelését jelentenék külföldi útjaink során...

Ez alá is rendelhető névvel ellátott célállomás (subdomain)

A második szintű domaintől ezt is ponttal választjuk el.

Ahhoz, hogy a két különböző címzési rendszer egymásnak megfeleltethető legyen, DNS-eket (Domain Name Server) alkalmazunk a hálózatban. Magán az interneten több ezer ilyen kiszolgáló található, de belső hálózatunk számára magunk is létre hozhatunk ilyet.

A már említett UDP protokollt használják rendszerint a névfeloldási kérések csomagjainak küldésére.

Routing Information Protokoll (RIP)

Routing Information Protokoll v1 (RIPv1)

A RIP a XEROX PARC által kifejlesztett GWINFO nevû protokollból származik, melyet az XNS-be RIP néven integráltak. Az Internethez 1982-ben kapcsolódik, amikor a BSD UNIX egy „routed" elnevezésû RIP implementációval került forgalomba, mely segítségével a munkaállomások route-olhattak. Elôször az XNS Internet Transport Protocols nevû publikációban 1981-ben, majd 1988-ban az RFC1058-ban jelent meg formális specifikációja. 1993-ban az RFC1387-88-ban a RIP második verziója is napvilágot látott, sok ésszerû javítással, bár ekkor az OSPF már végleges formájában létezett. A RIP egyszerûsége miatt a mai napig legnépszerûbb Internetes IGP protokoll, bár idôközben több hiányosságára fény derült. Számos más routing protokollnak szolgál alapjául, például az AppleTalk RTMP-nek (Routing Table Maintenance Protocol), de a Novell, a 3Com és a Banyan is használt RIP származékokat. Mi a RIP IP verzióját tekintjük át.

A RIP alapvetôen egy lapos, egyutas, distance-vector protokoll. A RIP-et futtató router-ben konfigurálni kell az interface-eire kapcsolt hálózatok (link-ek) címeit és egy csomagnak az adott link-en való átküldésének költségeit, valamint az idôzítéshez használt idôértékeket. A célok, amikhez vezetô utakat a RIP nyilvántart, lehetnek hálózatok, subnet-ek, állomások, vagy a default router. Azt, hogy egy cím subnet vagy állomás, csak a subnet maszk segítségével lehetne eldönteni, ami viszont a RIPv1 feltételezése szerint csak az adott hálózaton belül áll rendelkezésre. Éppen ezért a hálózat határain kívülre nem szabad a hálózat belsô subnet-eit célként hirdetni, csak az egész hálózatot (subnet hiding). Hasonlóképpen egyedi állomások hirdetése sem célszerû, csak a router-ek közötti kommunikációt növeli.

A RIP egy célponthoz táblázatában a következô információkat tárolja:

1. A célpont IP címét (0.0.0.0 a default route címe)
2. Az odavezetô út költsége, a 16-os költség a „végtelent", az elérhetetlen célpontot jelöli.
3. Az odavezetô út elsô router-e
4. Idôzítôk

A default route egy teljesen közönséges cél, amit a default router-ek hirdetnek 0 költséggel, akár többen is. Így minden router a hozzá legközelebbi default router felé irányítja az ismeretlen csomagokat.

A frissítô üzeneteket 30 másodpercenként küldik a router-ek, kis varianciával, hogy elkerüljük a szinkronizációt, azaz azt, hogy minden router egyszerre küldje frissítô üzeneteit 30 másodpercenként nagy tumultust okozva ezzel a link-en. Minden bejegyzéshez két idôzítô tartozik, az egyik a timeout, a másik a szemétgyûjtés ideje. A timeout idôzítô méri a bejegyzés utolsó frissítése óta eltelt idôt. Ha egy útvonal végtelen költségûvé válik, vagy 180 másodpercig semmiféle információ nem érkezik róla, akkor végtelenre állítjuk, azonnal

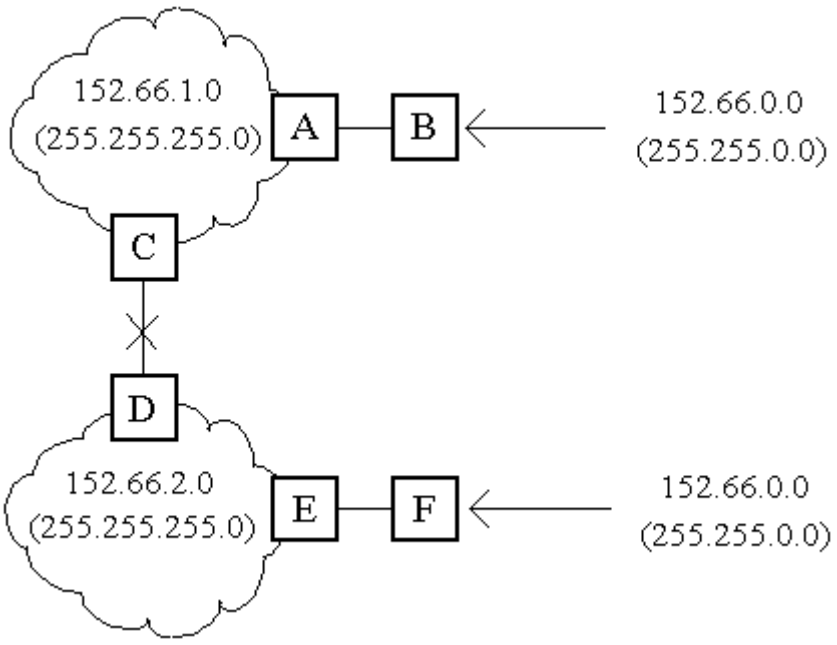
szétküldünk egy üzenetet, hogy megváltozott a bejegyzés költsége és elindítjuk 120 másodpercrôl a szemétgyûjtô idôzítôt. Ha ebben az idôszakban sem változik a bejegyzés állapota, töröljük. A protokoll és split horizon, a poisoned reverse és a triggered update eszközeivel, melyeket az elôzô részben ismertettünk.

A RIP meglehetôsen robosztus és magán viseli a distance-vector protokollok két fô jellemzôjét, az egyszerûséget és a nem túl gyors konvergenciát (topológiaváltozás esetén a végtelenig számolás miatt eltelhet egy kis idô, mire stabilizálódik a router-ek állapota). A 16-os végtelen érték miatt, ha minden link költsége is csupán 1 (ami a tipikus), akkor is csak meglehetôsen kis hálózatokban használható. Ez azonban nem is baj, lévén, hogy az algoritmus nem igazán nagy hálózatokon optimális.

Routing Information Protokoll v2 (RIPv2)

A RIP csomagformátumában számos tartalék mezô volt, melyek értékét kötelezôen 0-ra kell állítani. Ezeket a mezôket felhasználva az eredeti protokoll bôvítésére nyílt mód, alapvetôen két újítás került bele.

Az egyik az, hogy a célpontok címe mellett most a subnet maszkot is tartalmazza a frissítô üzenet, még a hálózat határán kívül is. Erre azért volt szükség, mert ha a hálózat belül kettészakadt úgy, hogy mindkét résznek volt kapcsolata a külvilággal, akkor bár fizikailag minden subnet elérhetô mégis elôfordulhat, hogy kívülrôl nem jut el csomag valamelyikbe. Minthogy kifelé csak a teljes hálózatot hirdetjük, csupán a véletlen mûve, hogy a 152.66.1.0 subnet-be címzett csomag a B vagy az F router-en keresztül érkezik. Ha a B-n, akkor célbatalál, ha az F-en akkor nem. Ha a subnet-ek maszkjait kívülre is terjeszthetjük, akkor az F router csak a rajta keresztül valóban elérhetô 152.66.2.0 subnet-et terjeszti kifelé és felé nem irányulnak a másik subnet-be címzett csomagok. Ezen felül lehetôvé vált változó hosszúságú subnet-ek alkalmazása is, errôl bôvebben a CIDR fejezetben olvashatunk.



A másik bôvítés a RIP csomagok hitelesítése. A RIPv1 mûködését könnyen megzavarhatja bárki, aki képes az 520-as UDP porton adni és venni, mert így magát egy RIP-et futtató router-nek álcázhatja. Példának okáért bármilyen 0 költségû utat terjeszthet, magára irányítva ezzel a forgalmat. A RIP csomagok éppen ezért hitelesítô információt hordozhatnak. Egy 16 bites mezô határozza meg a hitelesítési algoritmus típusát és 16 byte-nyi hitelesítô információ átvitelére van lehetôség minden RIP csomagban. Jelenleg csak az egyszerû jelszavas hitelesítô algoritmus definiált, amit nem túl nehéz feltörni, a jelszót kivéve a csomagból és más csomagba átmásolva máris hitelesített csomagokat küldhetünk. Olyan algoritmus, ami csak 16 byte-ot használ, idôfüggô (a csomagok letárolása és késôbbi újraküldése ellen) és biztonságos éppen kidolgozás alatt áll.

A RIP további bôvítési lehetôségei

A RIP (és általában a distance-vector protokollok) további finomítási lehetôségekkel rendelkeznek. Ezek közül az egyik a rendszeres frissítô üzenetek között beálló szinkronizáció megtörésére vonatkozik. Bár az üzeneteket nem pontosan 30 másodpercenként küldjük, hanem ettôl egy kis, véletlen értékkel eltérô idôközönként, mégis kialakulhat a szinkronizáció, mégpedig a következôképpen. Mikor egy router-ben lejár a 30 másodperces idôzítô, nekiáll összeállítani a frissítô üzenetét. Ha eközben más router-ektôl frissítô üzenetet kap, azokat elôbb feldolgozza, majd visszatér félbehagyott munkájához. Ha elküldte üzenetét, akkor újra beállítja idôzítôjét. Így a 2 frissítô üzenet elküldése közötti idô nem 30 másodperc, hanem 30 másodperc plusz a feldolgozásra fordított idô. Ha néhány router szinkronizálódott, akkor egymás után küldik el üzeneteiket. Ha valamely másik router pont ezen elküldési idôszak alatt állítja össze üzenetét, máris szinkronizálódott. Minél több router van együtt, annál hosszabb a feldolgozási idô és annál inkább valószínû, hogy egy újabb router beleesik a már szinkronizálódott csoportba, azon router-ek közé, akik szinte egyidôben adják és veszik frissítô üzeneteiket. Erre megoldás, ha a frissítési idô szélesebb skálán 15 és 45 másodperc között változik véletlenszerûen. Ez elég nagy variancia a már kialakult szinkron csoportok megtöréséhez is.

Másik javítási lehetôség adódik a 30 másodpercenkénti frissítô üzenetek elhagyásával. Így különösen stabil hálózatok esetén jelentôsen csökkenthetô a routing protokoll által generált forgalom. Különösen ott elônyös ez, ahol „hallgatni arany", a kapcsolatorientált hálózatok felett, ahol a (különösen a ritkán elküldött) csomag postázása elôtt kapcsolatfelépítés szükséges. Ha nem küldünk periodikus frissítést, akkor viszont nyugtázni kell a topológiaváltozás hatására elküldött frissítô üzeneteket. Akkor azonban, ha azt halljuk, hogy egy hálózat az eddigi irányban elérhetetlen, semmiféle lehetôségünk nincs arra, hogy kitaláljuk, vajon más irányba elérhetô-e. Eddig ilyenkor vártunk a periodikus frissítô üzenetekre és azokban kerestünk a végtelennél olcsóbb utat. Most viszont vagy minden szomszédunk teljes „útajlánlatát" le kell tárolni, vagy magunknak kell lekérdezni azt a célpontot, amire szükségünk van. Az egyik sok memóriát igényel, a másik pedig minden szomszéd megszólítását, ami a hallgatni arany, beszélni egy egység a telefonkártyáról elv alapján elég drága lehet.

A RIP által használt primitív költség-rendszert is javíthatjuk összetett költségek bevezetésével. Az IETF SIP munkacsoportja kidolgozott egy ilyen összetett költséget. A költség egyik eleme lehetne a hop-számláló, ami jelezheti a végtelen értéket, a másik pedig az adott link sávszélessége. Mikor egy újabb szegmenst adunk az útvonalhoz, a hop-számlálót egyel növeljük (csakúgy mint a RIP-ben, ha 1 a link költsége), a sávszélességek közül pedig a kisebbet vesszük. A hosszú útvonalak kiküszöbölése érdekében a kapott sávszélességet még mintegy 20%-kal csökkentjük. A választás kritériuma a sávszélesség, azonos sávszélességek esetén pedig a hop-számláló.

Mindezen csiszolások azonban nem érintik a distance-vetcor protokollok fô problémáját, a hurkok kialakulását. Erre a source-tracking (forrás-nyomonkövetés) módszere jelenthet megoldást. Lényege, hogy a routing táblázatokban nemcsak a célpont címét és a költséget tüntetjük fel, hanem az odavezetô úton a célponthoz legközelebb esô router címét is. Ha bejegyzésünket terjesztjük, akkor a legközelebbi router címét változtatás nélkül továbbadjuk. Ilyen módon minden célponthoz képesek vagyunk rekonstruálni az útvonalat, ha minden router célként szerepel táblázatunkban. B szemszögébôl például az X célponthoz E a legközelebbi router, ahhoz A, A-hoz pedig B maga. Ha kialakulna az A, B, C hurok, akkor ez is hamar kiderülne.

A vázolt algortimus igen elegáns, bár nem könnyû a RIP-be integrálni. Elôször is új mezôket kellene bevezetni a frissítô üzenetekbe a legközelebbi router címének továbbadására. Másodszor egy router-nek több címe van, minden interface-hez egy-egy. Vajon melyiket használjuk? Másfelôl viszont feltételezzük, hogy a fenti példát alapul véve B-ben minden router-hez van bejegyzésünk (nevezetesen E-hez és A-hoz), ami alapján összeállíthatjuk az útvonalat. Ez nem igaz a RIP-ben, ahol legjobb esetben is minden link-hez (subnet) van bejegyzésünk. A router-ek címét tehát párosítani kellene a link-ek címeivel.

Az említett nehézségek mind leküzdhetôek lennének, az érzékelt hurkokat a költség végtelenre állításával azonnal feloldhatnánk, mégis általános a vélemény, hogy a megoldás ellentétben van a RIP egyszerû implementálhatóságával és jelentôsen megnövelné a szükséges számítási teljesítményt.

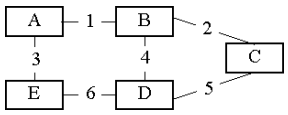
Open Shortest Path First (OSPF)

Az OSPF (Open Shortest Path First) egy link-state protokoll, melyet az IETF Interior Gateway Protocol munkacsoportja fejlesztett ki elsôsorban a RIP hiányosságai miatt. 2. verziója az RFC1247-ben 1991 júliusában jelent meg és körülbelül ötször olyan terjedelmes, mint a RIP leírása. Valóban az OSPF bonyolult, ám sokkal kifinomultabb, kevesebb sávszélességet foglal, hurokmentes és számos más elônnyel rendelkezik a RIP-hez képest.

A link-state protokollok mûködése két részbôl áll. Elôször minden állomás felderíti a hálózat topológiáját, majd a kapott gráfban megkeresi a legrövidebb útvonalat és az ahhoz tartozó elsô állomást, amely felé továbbítani fogja a csomagot. Nyilvánvaló, hogy életbevágóan fontos, hogy a router-ekben levô topológia egyezô legyen és a legrövidebb út kiszámítása is mindenhol ugyanazon algoritmus szerint zajoljon, különben teljes káosz alakul ki. (Az A router B felé számítja a legrövidebb útvonalat, a B meg A felé és kész a galiba.) Az utóbbi

feltétel könnyen teljesíthetô, ám a topológiai adatbázisok szinkronizálása komoly munkát igényel.

A hálózat topológiáját a link-ek állapotát leíró rekordok (link-state records) terjesztésével tudatják egymással az állomások. Az egyszerûség kedvéért egyenlôre ne tegyünk különbséget az állomások és a router-ek között.



Második példa-hálózat

A fenti egyszerû, hálózat topológiai adatbázisa 12 rekordból áll, minden link-hez kettôbôl, hisz a link mindkét végén levô állomás létrehoz egy rekordot. A rekordokban szerepel a két állomás, ami között a link fut, a link száma és költsége is.

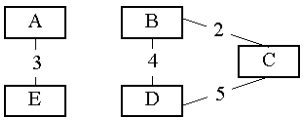
Honnan	Hova	Link	Költség
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

A második példa-hálózat topológiai adatbázisa

A link-ek állapotát leíró rekordokat idôbélyeggel látják el a router-ek, majd minden irányban terjeszteni kezdik. A kapott rekordokat feljegyzik saját topológiai adatbázisukba, az idôbélyeggel együtt, majd továbbadják. Ha olyan rekordot hallanak, amely már szerepel az adatbázisban és a hallott verziónál régebbi, akkor figyelmen kívül hagyják. Ezzel megakadályozzák, hogy egy-egy rekord örökké keringjen a hálózatban.

Ha egy link meghibásodik, errôl a két végén levô állomás értesül, és mindketten körbeadnak egy üzenetet, melyben jelzik, hogy a kérdéses link költsége végtelenre módosult, errôl mindenki értesül és a topológiai adatbázisok szinkronban maradnak.

Ha a hálózat kettészakad, a két rész képtelen egymást értesíteni a további változásokról. Példánkban az 1. és a 6. link meghibásodása után az A és E állomások már nem értesülhetnek a 2., 4. vagy 5. link hibájáról. Ennek természetesen semmiféle káros következménye nincs, minthogy az 1. és 6. link-ek költsége végtelen, így az A és E számára a B, C és D állomások úgyis elérhetetlenek, lényegtelen a köztük lévô link-ek állapota.



A második példa-hálózat kettészakadt állapotában

Más a helyzet azonban, ha az 1. link újra mûködôképes lesz. Errôl ugyan körbeküldenek az A és B router-ek egy üzenetet, azonban ez nem elég, hiszen ha idôközben például a 2. link állapota módosult, akkor A-ban errôl még a régi információ található. Éppen ezért az A és B állomások szinkronizálják adatbázisaikat. Ez azonban még mindig nem elég, E miatt, akivel szintén szinkronizálni kell. Megállapíthatjuk, hogy a topológiai változáskor körbeadott üzenet nem elégséges, ennél erôsebb szinkronizáció kell, amely célszerûen párokban zajlik.

A link-state módszernek több elônye is van a distance-vector protokollokkal szemben.

Egyrészt topológiai változás esetén a konvergencia gyors, hisz kis számú, rövid rekordot kell csak körbeadni a hálózatban, míg a distance-vector algoritmusok esetén egy link hibája számos célpontot érinthet és ez hosszú frissítô üzeneteket eredményezhet. Ráadásul a link-state algoritmus konvergenciája közben nem alakulhatnak ki hurkok.

Másrészt itt a költségek egyszerûbben tehetôk összetetté. Míg a distance-vetcor esetén a végtelen értékét alacsonyan kellett tartani, itt erre nincs szükség.

Harmadrészt a topológiai adatbázisból nem csupán egy, de több útvonal kiszámolható, melyek között megoszthatjuk a forgalmat, ez nem igényel további kommunikációt vagy memóriát, míg az EIGRP-ben levô multipath lehetôségek kihasználásához összes szomszédunk által terjesztett bejegyzéseket tárolni kell.

A multipath routing alatt két lehetôséget értük. Az egyik esetben csak akkor osztjuk meg a forgalmat több útvonal között, ha holtversenyben a legolcsóbbak. A második esetben a forgalom egy részét olyan útvonalra engedjük, amelyik nem a legolcsóbb, de még elfogadható. Mindkét megoldás esetén kisebb lesz a csomagok késleltetésének ingadozása, a több útvonal miatt az effektív sávszélesség is nagyobb és az egyik például a legolcsóbb útvonal kiesése esetén a forgalom mintázata nem annyira ugrásszerûen változik meg, hisz a csomagok egy része eddig is más útvonalon haladt.

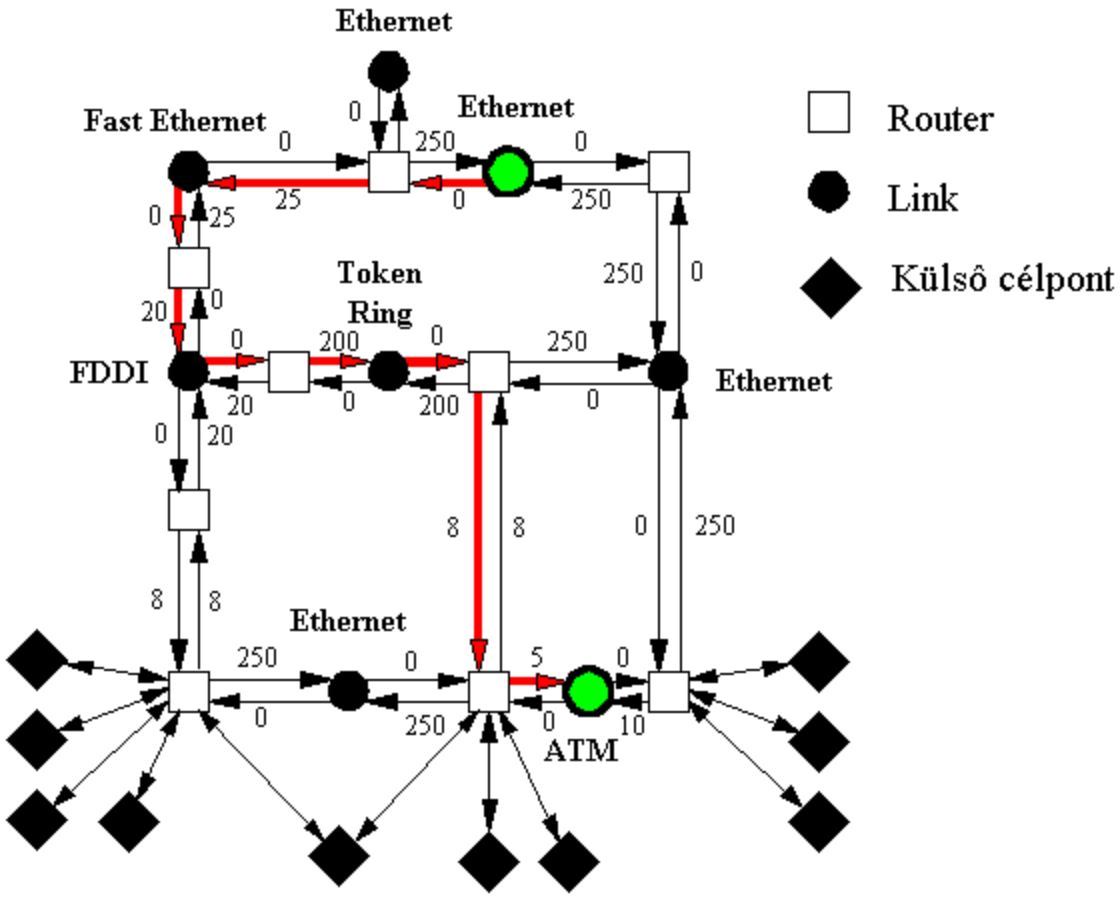
Ha azonban nem csak a legolcsóbb utakat használjuk, hurkot hozhatunk létre a hálózatban. A fenti példában E állomástól C felé 2 útvonalat tekinthetünk. Az egyik a közvetlen és ennél kétszer drágább a B-n át vezetô. Ilyenkor a forgalmat valamilyen arányban, például kétharmad, egyharmad érdemes megosztani. B állomáson hasonló a helyzet, ott is két útvonal van, a közvetlen és a E-n át vezetô. A forgalmat ott is ilyen módon megosztva, annak

egy része (az E felé irányított egyharmad harmada) vissza fog érkezni B-be. Ennek a visszaérkezett forgalomnak harmada ismét E felé távozik majd, a folyamat a végtelenségig tart. Igaz, hogy nagyon hosszú ideig csupán a forgalom igen kis százaléka kering a két állomás között, mégis a 4. link igen hamar torlódásossá válhat. A problémán úgy segíthetünk, ha csak olyan állomás felé továbbítunk csomagot, amelyik közelebb van a célhoz, mint mi, így attól nem kaphatunk vissza csomagot. A gyakorlatban azonban inkább csak a legjobb, de egyenlô költségû utak között szokás forgalmat megosztani, ez az EIGRP alapbeállítása is.

Maga az OSPF elnevezés onnan ered, hogy a kialakult topológiai gráfban a legrövidebb utat a Dijkstra nevéhez fûzôdô „legrövidebb utat elôre" (shortest path first, SPF) algoritmus szerint keresik a router-ek [5]. Ez egy igen hatékony O(N*log N) rendû algoritmus, ahol N a link-ek száma és ennyi idô alatt a gráfból az összes célponthoz meghatározza a legrövidebb utat.

Az OSPF elkülöníti az állomásokat és router-eket, hiszen csak az elôbbieken fut routing protokoll. Minthogy az egy link-en levô állomások teljesen egyformák a route-olás szempontjából azokat fölösleges megkülönböztetni, elég a link-et tekinteni. Az OSPF topológiai adatbázisában tehát nem állomások és router-ek közötti kapcsolatok, hanem router-ek és link-ek közötti kapcsolatok szerepelnek, a link-ek jelképezik az összes rájuk kapcsolt állomást. Az OSPF háromféle link-et különböztet meg.

1. Pont-pont link-ek 2 router között
2. Broadcast jellegû link-ek, mint például Ethernet vagy FDDI, ahol egy csomaggal minden állomáshoz információt juttathatunk el.
3. Nem broadcast jellegû linkek, mint például X.25 vagy ATM.



OSPF topológiai gráf

A pont-pont link-ek nem jelentkeznek csúcsként az adatbázisban, hiszen ezeken a link-eken nincs állomás, így csomagok célpontjai sem lehetnek.

A router-ek felelôsek a saját link-state rekordjaik terjesztéséért, a link-ek link-state rekordjait pedig a link egyik kiválasztott router-e juttatja a hálózatba, ezeknek a rekordoknak a költsége mindig 0. A kiválasztott router meghatározása a Hello protokoll szerint, automatikusan történik. A külsô célpontok információit a hálózat szélén levô router-ek terjesztik.

Ha egy link-en több router is van, nem szükséges mindegyiknek, mindegyikkel szinkronizálnia adatbázisát, ez túl sok fölösleges kommunikációval járna. Ehelyett mindegyik csupán a kiválasztott router-rel ellenôrzi a szinkront és csak azzal szinkronizál eltérés esetén; könnyen belátható, hogy így bármely két router azonos adatbázisra fog eljutni. A kiválasztott router ilyenformán kitüntetett szerepet játszik, vele szinkronizálják adatbázisukat a többiek és ô a felelôs a link állapotát leíró rekord terjesztéséért. A kiválasztott router meghibásodása esetén új kiválasztott router-t kell választani és azzal minden router-nek szinkronizálni az adatbázisát. Ez a folyamat meglehetôsen hosszú idôt vesz igénybe, éppen ezért a kiválasztott router mellett tartalékot is választanak és nemcsak a kiválasztott, de a tartalék router-rel is szinkronizálják adatbázisukat. Így a kiválasztott kiesése esetén a tartalék azonnal a helyére léphet, a hálózat mûködése nem szünetel, mialatt új tartalékot választanak és azzal összeszinkronizálódnak.

Az OSPF 3 alprotokollból áll.

1. A Hello protokoll, ami segítségével a router-ek a link-ek állapotát tesztelik, felderítik egymást és meghatározzák a kiválasztott router-t.
2. Az Exchange protokoll, ami segítségével topológiai adatbázisok szinkronizációja folyik.
3. A Flooding protokoll, ami a link-state rekordok terjesztéséért felelôs.

Mikor egy hálózatot üzembe helyezünk, a router-ek Hello csomagokat küldözgetnek egymásnak, melyben felsorolják azokat a router-eket, akikrôl ôk tudnak az adott link-en. Ily módon minden router azt is ellenôrizheti, hogy róla kik tudnak. A nem broadcast jellegû link-eken szükséges minden router-t a többi router címével konfigurálni, broadcast jellegû hálózaton a Hello protokoll ezt képes felderíteni.

A Hello csomagokban minden router egy elôre beállított prioritást is közzétesz magáról, a legnagyobb prioritású router lesz majd a kiválasztott. Azonban ha a kiválasztás után egy még nagyobb prioritású router kapcsolódik a link-re, a kiválasztás marad, mert a kiválasztott router változása költséges a sok újraszinkronizáció miatt. A frissen bekapcsolt router mindössze a kiválasztottal és a tartalékkal szinkronizálódik össze.

Maga az szinkronizáció az Exchange protokoll segítségével zajlik. Teljesen fölösleges lenne a teljes topológiai adatbázist lecserélni, hiszen a legtöbb esetben csak kis eltérések vannak. Éppen ezért elôször mindkét router leíró csomagokban közli a másikkal milyen bejegyzések milyen idôbélyeggel szerepelnek az adatbázisában. Ebbôl mindkét router kitalálhatja, hogy mely bejegyzések frissebbek a másikban és azokat felsorolja kérô csomagok formájában. A másik router pedig közönséges link-state rekordokkal válaszol, melyeket a vevô a Flooding protokoll szerint tovább terjeszt a hálózatban, mintha közönséges link-state frissítô rekordok lennének.

A Flooding protokoll terjeszti a link-state rekordokat, mûködése egyszerû: ha a vett rekordok frissebbek a mi adatbázisunkban szereplôknél, akkor módosítjuk az adatbázist és tovább terjesztjük a rekordokat. Ha nem, nem csinálunk semmit.

Bár eddig az egyszerûség kedvéért sorrendiséget követtünk, a valóságban a három protokoll egyidôben mûködik. Tehát egyszerre zajlik:

1. a link meglétének vizsgálata
2. ezzel együtt a kiválasztott router mûködésének ellenôrzése
3. a szinkronizáció a kiválasztott és a többi router között, vagyis periodikusan sor kerül a leíró csomagok elküldésére, amelyekbôl kiderül, ha az adatbázisok nincsenek szinkronban
4. a kiderült szinkronizációs hiányok nyomában a kérô csomagok elküldése
5. a szinkronizáció miatt feltett kérdésekre adott válaszok, valamint a Flooding protokoll által terjesztett rekordok adása-vétele.

Az OSPF minden kommunikációja nyugtázott, az idôben befutó nyugta hiányában a csomagokat újraküldik, emiatt elôfordulhat, hogy a szinkronizációs folyamat elhúzódik és a

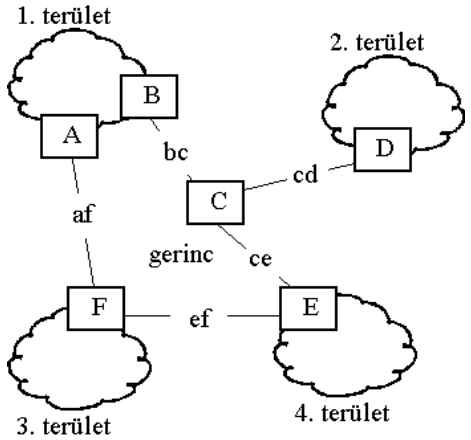
közepén teljesen máshonnan egy link-state rekord érkezik. Ez nem befolyásolja a szinkronizációt, még ha a szinkronizációban frissítendô rekordot érint is, ha a rekord újabb, mint az ismert, akkor a szinkronizációtól függetlenül terjesztjük, még szinkronizációs partnerünknek is.

A kommunikáció ezenfelül hitelesített is, bár jelenleg csak nagyon egyszerû jelszavas hitelesítés definiált, mint a RIPv2-ben amit feltörni könnyû, ám a késôbbiekben bonyolultabb eljárások is integrálhatók a protokollba.

A régi bejegyzéseket célszerû eltávolítani a topológiai adatbázisokból. Ezt azonban a szigorú szinkronizációs követelmények miatt egyszerre kell minden router-nek megtennie. Éppen ezért minden rekord életkort is tartalmaz, ami 0-ról indulva másodpercenként nô és a rekord terjesztésekor ezt is továbbadjuk. Ha eléri élettartama végét egy órát, törlôdik az adatbázisból, de errôl a Flooding protokoll segítségével az egész hálózatot tájékoztatják, így minden router törölheti a rekordot.

Természetesen ha az adott rekord megfelel a valóságnak, akkor a kiöregedés elôtt meg kell újítani. Éppen ezért minden rekordot a Flooding protokoll segítségével, ha valamilyen más okból nem kell hamarabb, akkor 30 perc elteltével mindenképpen megismétlünk.

Nagyméretû hálózatok esetén a topológiai adatbázis mérete nagyobb lehet a kívánatosnál. Éppen ezért az AS-t több területre oszthatjuk fel. A Flooding protokoll csak a területek belsejében terjeszti a link-state rekordokat, a topológiai adatbázis csak egy területen belül egységes és csak a terület belsejének térképét tartalmazza. A területek egy kitüntetett területen keresztül érhetik el egymást, ez a gerinc. Minden területnek muszáj legalább egy ponton a gerincre kapcsolódnia. A területhatáron lévô router-ek összegzik a területükön található link-ekig vezetô utak költségét és ezt az összegzô információt terjesztik a gerincre. A gerinc többi router-e ezeket a gerincre küldött összefoglaló információkat továbbterjeszti a saját területére, de úgy, hogy hozzáadja a gerincen való áthaladás költségét is.



OSPF területek és a gerinc

A fenti ábrán például a B router összegzi az 1. területen található összes link tôle való távolságát és ezt az információt a gerinc-területen belül a Flooding protokoll segítségével

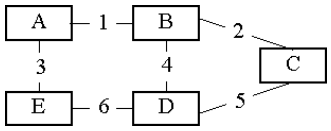
elterjeszti. Az E router az így megismert költségekhez hozzáadja a bc és a ce link-ek költségét és így terjeszti a 4. terület belsejében a Flooding protokoll segítségével. Ugyanezt az információt megkapja az A router-tôl is az af és ef link-eken át, ebben az esetben az 1. terület minden célpontjához 2 útvonal áll rendelkezésre, az egyik az A, a másik a B router-en át. Itt a distance-vector eljáráshoz hasonlóan a kisebbet választjuk és terjesztjük. Nem áll fenn azonban a hurok kialakulásának veszélye, hisz a területeken és magán a gerincben is egy link-state algoritmus választ utat, a distance-vector elem pedig egy teljesen hierarchizált helyzetben lép be: a csomag föllép egy területrôl a gerincre, majd onnan a cél-területbe lép. Itt hurok ki nem alakulhat.

Ha egy terület például az 1. kettészakad, de mindkét része kapcsolatban marad a gerinccel, akkor a topológiai adatbázisok szinkronizációja után a két router csak azokat a célpontokat fogja terjeszteni a gerincre, melyek az ô felében vannak. Így elkerülhetô az, hogy a területnek szóló csomagok rossz félbe kerülnek és így nem jutnak el a címzetthez. Ez a probléma már a RIPv1-nél fölmerült a subnet hiding módszerével kapcsolatban. Ha a gerinc szakad szét úgy, hogy egy területen keresztül megmarad még az összeköttetés (például, ha az ef link szakad meg, akkor az 1. területen keresztül képes még kommunikálni a 3. és a 4. terület), akkor az OSPF alaphelyzetben nem képes fenntartani a konnektivitást. Ezen a problémán segítendô létrehozhatunk az A és B router-ek között egy virtuális link-et, ami az 1. területen át vezet. Az ilyen link költsége kiadódik az 1. terület topológiájából. Ezt a link-et az A és B router link-state rekordban a Flooding protokoll segítségével terjeszteni fogja a gerincben, így ha az ef link megszakad, a gerinc összeköttetése megmarad, ezen a virtuális link-en keresztül. A gerincen átküldött csomagok (például E-tôl F-nek) pedig valójában elhagyják a gerincet. A virtuális link azonban nemcsak hiba esetén kerülhet használatba, hanem akkor is, ha olcsóbb, mint a gerinc fizikai link-jei. Például egy D-tôl F-be küldött csomag haladhat a cd, bc, virtuális link, af útvonalon, ha ez olcsóbb, mint a cd, ce, ef. Ez persze felettébb valószínûtlen, hisz a gerinc link-jei valószínûleg gyorsabbak, mint a területekéi, hisz a virtuális link fizikailag több link egymásutánja lehet.

Az OSPF összetett költséget használ, 4 összetevôvel. Ezek:

1. A link sebessége
2. A link megbízhatósága
3. A link költsége (pénzben)
4. A link késleltetése

Ezek alapján több topológiai adatbázis építhetô fel, egy-egy a különbözô költségekhez és egy az „átlag" költséghez. A különbözô gráfokban számolt legrövidebb út adja rendre a leggyorsabb, legmegbízhatóbb, legolcsóbb és legkisebb késleltetésû útvonalat. Így lehetôség adódik az IP csomagokban levô TOS mezô figyelembevétele. Ha a csomagot valamelyik TOS bit beállításával adták fel, akkor a megfelelô gráfból kinyert legrövidebb (tehát legmegbízhatóbb, legkisebb késleltetésû, vagy legnagyobb sávszélességû) úton továbbítjuk, ha egyiket sem, akkor az „átlagos" költség alapján felépített gráfból származtatott úton. Ehhez azonban 5 topológiai adatbázis és 5 routing táblázat kell, ami drágítja a router-eket, így nyitott a lehetôség, hogy egy router csak a 0 TOS-t támogassa és csak egy (átlagos) topológiai adatbázist építsen fel.



Második példa-hálózat

Fontos, hogy a nem 0 TOS továbbítási útvonalon minden router támogassa a TOS-t, hiszen ha a fenti ábrában C-bôl B-be a nem 0 TOS (mondjuk a legnagyobb sávszélességû) útvonal a D-n keresztül vezet, D-bôl C-be pedig a 0 TOS útvonal C-n keresztül és D nem támogatja a TOS routing-ot, akkor C egy nagy sávszélességet igénylô csomagot D-nek ad fel, az viszont nem ismervén a sávszélességek alapján felépített topológiát, az átlagos költség alapján visszaküldi C-nek. A nem 0 TOS topológiai adatbázisokba tehát csak TOS router-ek rekordjait vehetjük fel. Ha az ily módon kapott topológia nem egybefüggô, mert túl kevés router támogatja a TOS-t, akkor a TOS alapján való route-olás csak az egybefüggô részeken belül lehetséges.

A területek közti összefoglaló információk terjesztéséhez hasonló módon terjeszthetünk külsô, az AS-en kívüli célpontokat is, melyekrôl egy EGP protokoll útján szerezhetünk tudomást. Az ilyen utak költségeként megadható egy második (type 2) költségfajta is, ami minden belsô használatú költségnél drágább. Ilyen esetekben az AS-en belüli továbbítás költsége nem számít, annál a router-nél fog távozni a csomag, amely a legolcsóbb type 2 költséget hirdeti, még ha az AS túlsó felén van is. Ez azért hasznos, mert az AS-ek közötti forgalom sokszor politikai megfontolásokat is követ, melyek fontosabbak az AS-en belüli költségeknél.

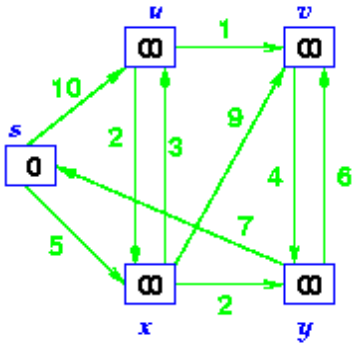
Összesen tehát négyféle információt terjeszt a Flooding protokoll:

1. A router-ek link-state rekordjait, ezek a router interface-eit és az azokhoz kapcsolt link-eket, azok költségeit, élettartamát és idôbélyegét, sorolják fel. A topológiai ábrán ezek a rekordok felelnek meg a router-ekbôl a link-ekbe mutató nyilakkal.
2. A link-ek link-state rekordjait, ezek a linken található router-eket sorolják fel. A topológiai ábrán ezek a rekordok felelnek meg a link-ekbôl a router-ekbe mutató nyilakkal, költségük mindig 0.
3. A területhatáron levô router-ek által terjesztett összefoglaló információkat. Ezek mindegyike egy célpontot (link) jelöl meg az odavezetô út költségével és azzal, hogy mely router hirdeti ekkora költséggel az utat.
4. A külsô utak, azok költsége és a hirdetô EGP router.

A külsô utak kezelése opcionálisan implementálható OSPF képesség, azon router-ek, melyek nem tudnak külsô utakat kezelni, az ismeretlen hálózatok felé a default route-n továbbítják csomagjaikat. Azon területeknek, melyek csupán egy router-en át kapcsolódnak a gerinchez, nincs szüksége sem a külsô célpontok, sem a más területen levô célpontok összegzô információira, hiszen úgyis minden a területen kívülre küldött csomag azon az egy router-en át távozik. Az ilyen vak (stub) területeken éppen ezért csak a default route terjesztésére van szükség.

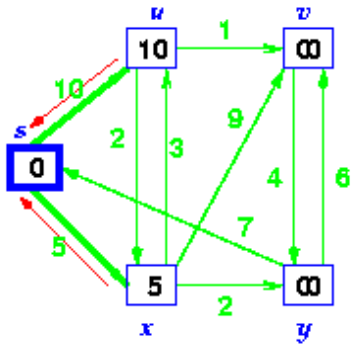
Az OSPF a link-eket az EIGRP-hez és a RIPv2-höz hasonlóan egy IP cím és egy maszk párosával definiálja, lehetôvé téve ezzel a változó hosszúságú subnet-ek használatát és a hálózatok aggregálását, errôl bôvebben a CIDR fejezetben olvashatunk.

Legrövidebb út keresése Dijkstra algoritmusával-példa



Kezdeti gráf adott, az s csúcsból szeretnénk megkapni a legrövidebb utakat minden csúcsba.

Minden csúcsnak végtelen a címkéje, kivéve amiből indulunk (ezt meg kell adni), ennek a 0 címkét adjuk.



Mindig az indulási csúcshoz legközelebbi csúcsot keressük. Ez az a csúcs, amihez vezető élsorozatban lévő élek címkéinek összege a legkisebb.Ezt az összeget a csúcshoz írjuk, ez lesz a csúcs címkéje.

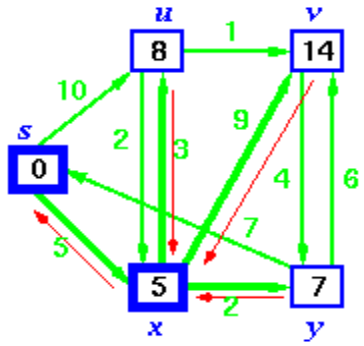
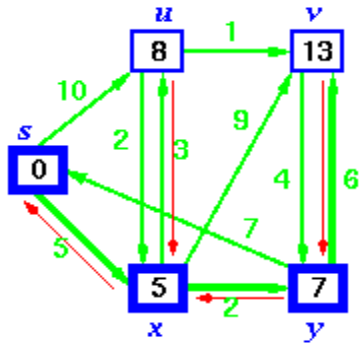
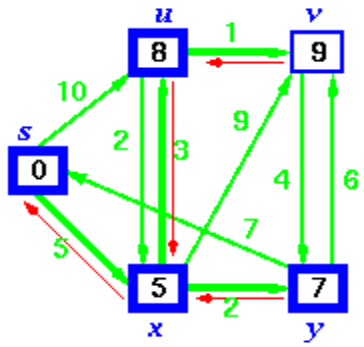
1. s címkéje 0, úgyhogy ezt választjuk. Karbantartunk egy halmazt, amiben azok a csúcsok szerepelnek, amikhez már tudjuk a hozzájuk vezető legrövidebb utat. Ezeket már tovább nem vizsgáljuk.

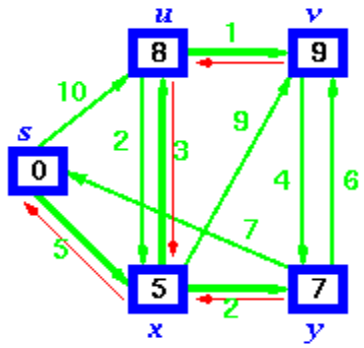
2. S={s}

3. Most az újonnan kiválasztott (S halmazba utolsónak bevett) csúcs, jelen esetben az s, összes szomszédját vizsgáljuk, és szükség esetén a CSÚCS címkéjét átírjuk. A szomszédok: u és x. Az u címkéje a végtelen helyett 10, x címkéje 5 lesz.

Minden lépésben ellenőrizzük az S-ben nem szereplő csúcsokat, és szükség esetén (amikor rövidebb utat kapunk) átcímkézzük őket.

4. Minden csúcsból indítunk egy pointert, ami abba a csúcsba mutat, amelyen át a rövidebb út vezet (ami miatt átcímkéztük) A pointereket az ábrán a legvékonyabb nyíl jelzi.

	1. Az indulási (s) csúcshoz legközelebbi csúcsot keressük, ami még nincsen S-ben, ez most x. **2. S={s}∪ {x}= {s, x}** 3. Most az újonnan kiválasztott (S halmazba utolsónak bevett) csúcs, jelen esetben az x, összes szomszédját vizsgáljuk, és szükség esetén a CSÚCS címkéjét átírjuk. A szomszédok: u, y, v. Az u címkéje a 10 helyett 8 lesz, hiszen x címkéjéhez egy 3 hosszúságú utat kell hozzáadni. Az y csúcs címkéje végtelen helyett 7 (5 + 2), a v csúcsé pedig 4 (5+9). 4. A pointereket beállítjuk az ábra szerint.
	1. Kiválasztjuk az s-hez legközelebbi csúcsot (a legkisebb címkéjűt), ami nincsen s-ben, ez y. **2. S={s}∪ {y}= {s, x, y}** 3. Most y szomszédjaihoz vezető utakat vizsgáljuk, ezek: x, s, v. AZ x-be vezető út 9 lenne, ez több, mint x jelenlegi címkéje, ezért nem javítjuk át x címkéjét. S-nek nulla a címkéje, ezt nem is kell vizsgálni. A v címkéje 14, de y-é 7, és y-ból egy 6 hosszúságú út vezet v-be, 7+6=13, ezért v címkéjét átállítjuk 13-ra. 4. Átállítjuk a v pointerét, hogy y-ra mutasson.
	1. Kiválasztjuk az s-hez legközelebbi csúcsot (a legkisebb címkéjűt), ami nincsen s-ben, ez u. 2. S={s}∪ {u}= {s, x, y,u} 3. Az u egyetlen szomszédja, ami még nincsen S-ben, a v. E csúcshoz u-n át 1 hosszúságú út vezet, tehát az u-n átvezető, s-ből induló, v-ben végződő út 9 hosszú, ezért a v címkéjét át kell javítani 9-re 4. a pointert y-ról u-ra kell irányítani..

	**2. S={s}∪ {v}= {s, x, y,u, v}** 1. (Kiválasztjuk az s-hez legközelebbi csúcsot (a legkisebb címkéjűt), ami nincsen s-ben, ez v.) 3. Mivel S-ben a gráf minden csúcsa szerepel, tovább nem vizsgálódunk. Az így kapott címkézett élű, csúcsú gráfban minden csúcs címkéje az s-ből a csúcshoz vezető legrövidebb út hossza-magát az utat a pointerekből kapjuk.

DIJKSTRA ALGORITMUS

Bemenet: súlyozott (irányított gráf), a súlyok pozitív számok

G csúcsai: V={v1, v2, …vn}

Súlyok: w(vi, vj), ez legyen végtelen, ha az él nincsen a gráfban.

Adott kezdőpont, s.

Kezdeti értékek beállítása

FOR i:=1 TO n

Hossz(vi):= végtelen /* hossz az odavezető út hossza és egyben a csúcs címkéje

Hossz(s):= 0 /* a kezdőpont címkéje 0

DO WHILE S=V /* addig amíg a gráf minden csúcsa benne lesz S-ben

u:= olyan csúcs, amelyre Hossz(u) minimális és nincsen S-ben

S:=S∪{u}

DO minden csúcsra ami nincsen S-ben:

IF hossz(u)+w(u,v)<hossz(v) THEN hossz(v):= hossz(u)+w(u,v)

/* itt módosítjuk a csúcsok címkéit

END FOR

Adminisztratív távolságok

Az útvonal forrása	Adminisztratív távolság	Alapértelmezett mérték(ek)
Csatlakoztatva	0	0
Statikus	1	0
EIGRP összevont útvonal	5	
Külső BGP	20	Rendszergazda által megadott érték
Belső EIGRP	90	Sávszélesség, késleltetés
IGRP	100	Sávszélesség, késleltetés
OSPF	110	A kapcsolat költsége (sávszélesség)
IS-IS	115	A kapcsolat költsége (rendszergazda által megadott érték)
RIP	120	Ugrásszám
Külső EIGRP	170	
Belső BGP	200	Rendszergazda által megadott érték

CSOMAGSZŰRÉS CISCO ROUTEREKEN ACL-EK SEGÍTSÉGÉVEL

Hozzáférés-vezérlési listák

Az Access Control List (ACL), azaz hozzáférés-vezérlési lista a forgalomszűrés egyik legelterjedtebb változata [2]. Az ACL-ek segítségével hozzáférés vezérlést biztosítunk egy erőforráshoz. Segítségükkel ellenőrizhetjük a hálózatba bejövő illetve kimenő forgalmat, és szükség esetén még szűrhetjük is azt. A forgalomszűrés javítja a hálózat teljesítményét.

Az ACL segítségével az elosztási rétegben korlátozható a hozzáférés, és megakadályozható a nem kívánt forgalom központi hálózatba jutása. A hozzáférési listával ellenőrizhető a forgalomirányító interfészein áthaladó hálózati forgalom. Ez azt jelenti, hogy az OSI modell 3. rétegében dolgozunk, vagyis megelőzzük jóval a szoftveres védelmet. Az ACL-ek engedélyezhetnek és tilthatnak is forgalmat a megfelelő szabályokkal. Az ACL-ek megadási sorrendben hajtódnak végre, a szoftver végigmegy szabályokon, és amelyik megfelelő neki, azt végrehajtja. Ha nincs a kérésre vonatkozó meghatározás, az egyéb beállítások lépnek érvénybe. Háromféle ACL típus különböztethetünk meg, ezek a normál, a kiterjesztett és a nevesített ACL.

Normál ACL

A normál ACL (Standard ACL) a legegyszerűbb a három típusból. Forrás IP-cím alapján végzi a szűrést, teljes protokollműködés alapján tiltja vagy engedélyezi a forgalmat. Ha egy ilyen ACL nem engedélyezi egy munkaállomás IP forgalmát, az erről az állomásról érkező összes szolgáltatást letiltja. Lehetőségünk van egy adott felhasználó vagy helyi hálózat számára engedélyezni az összes szolgáltatás elérését a forgalomirányítón keresztül, míg az összes többi IP-cím esetén tilthatjuk a hozzáférést. A normál ACL-ek a hozzájuk rendelt azonosítási szám alapján azonosíthatók be. Az azonosítási számnak 1 és 99, illetve 1300 és 1999 közé kell esnie.

Pl. a **`Router(config)#access list 2 permit host 172.16.1.80; ACL a 172.16.1.80`** IP címet engedélyezi.

Kiterjesztett ACL

A kiterjesztett ACL (Extended ACL) már nem csupán a forrás IP-cím alapján, hanem a cél IP-cím, a protokoll és a portszámok segítségével is szűrhet. Sokkal elterjedtebb, mint a normál ACL, mivel jobb ellenőrzést tesz lehetővé, és specifikusabb is. Azonosítási számuknak 100 és 199, illetve 2000 és 2699 közé kell esniük.

Pl. a **`Router(config)#access list 102 permit 192.168.2.0 0.0.0.255 any`**; ACL a 192.168.2.0 hálózat minden állomását engedélyezi, ugyanakkor minden mást tilt. Továbbá a **`Router(config)#access-list 103 deny tcp any 192.168.2.0 0.0.0.255 range 20 2`**; a teljes FTP forgalmat letiltja. A **`Router(config)#access-list 101 deny tcp 195.220.0.0 0.0.255.255 0.0.0.0 0.0.0.0 eq 80`**; ACL-lel tiltjuk a 195.220.0.0/16 hálózat felől a HTTP (80-as port) kéréseket bármilyen célhálózat felé.

Az ACL definiálását egy interfészhez történő hozzárendelés követi

`(config)#interface Serial 0`
`(config-if)#ip access-group 1 out` (kimenő interfész) (config)#interface Ethernet 0
`(config-if)#ip access-group 101 in` (bejövő interfész)

Nevesített ACL

A nevesített ACL (Named ACL, NACL): normál vagy kiterjesztett hozzáférési lista, ahol az azonosító szám helyett egy névvel hivatkozunk a listára. A nevesített ACL-ek beállításához a forgalomirányítón NACL üzemmódban kell lennünk.

Az ACL-ek végén mindig van egy implicit tiltás, tehát mindent tiltunk alapesetben, azon kívül, amit engedélyeztünk. Ez a gyakorlatban azért bevett szokás, mert ilyenkor nem a tiltásokat, hanem az engedélyeket vesszük sorra, amit a szükséges ACL-ekkel tudunk megadni. Ebben az esetben nem fordulhat elő az eset, hogy a biztonsági tervezés alatt elfelejtenénk bármely biztonsági szempontból fontos elem tiltását!

Ha a nemkívánatos forgalomforráshoz közel tiltunk, akkor a forgalom nem halad keresztül az egész hálózaton, és foglal le értékes erőforrásokat. A hozzáférési listák minden olyan forgalmat ellenőriznek az ACL-ben megadott szabályoknak megfelelően, amelyek áthaladnak az eszköz megadott interfészén. Az ACL-ek helytelen használatából előfordulható hibák az alábbi típusúak lehetnek.

- Az összes csomag ellenőrzése jelentősen leterheli a forgalomirányítót, így kevesebb időt tud fordítani a csomagtovábbításra. Ilyenkor használható a sorba állítás, amikor protokollok szerint a router egyes csomagokat előbbre vesz, és egyes csomagokat, amelyek nem fontosak, fel sem dolgoz.
- A rosszul megtervezett ACL-ek sokkal nagyobb terhelést okoznak, ami a hálózat működésében zavart, hibát okozhat.
- A nem megfelelően elhelyezett ACL-ekkel pont az ellenkezőjét érhetjük el, mint amit szerettünk volna. Blokkolhatjuk az engedélyezni kívánt, és engedélyezhetjük a tiltani kívánt forgalmat [9].

A jól megtervezett hozzáférési listákkal csökkenthetjük a hálózat terhelését, és jóval kisebb sávszélességet használunk fel.

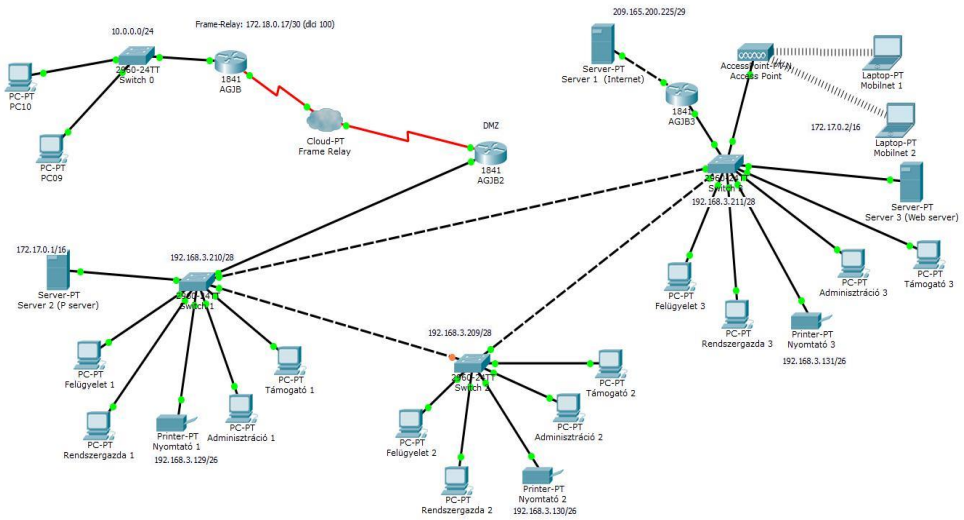
Konkrét megvalósított példa Cisco Packet Tracerben

Az alábbiakban röviden bemutatjuk a szimulációhoz használt Cisco Packet Tracer programot. A Cisco Packet Tracer egy emulációs program, amiben könnyen megtervezhetjük, létrehozhatjuk, és tesztelhetjük a kigondolt hálózati konfigurációt. A programban megjelenő eszközöket konzolosan vagy varázsló segítségével állíthatjuk be. Számos forgalomirányítót, kapcsolót, vezeték nélküli eszköz és végberendezést tartalmaz a program, ezeket a megfelelő beállítások után a valóságnak megfelelően működtethetjük.

Konkrét példánkban egy kitalált cég hálózatát hoztuk létre, melynek feladataihoz a tartozik többek között hivatalos esemény rögzítése, megosztása élőben, illetve későbbi időpontban ezek webkiszolgálóról történő megtekintése. Ezen kívül a cég munkatársai felügyelik a szervereken tárolt videó anyagokat, és esetleges probléma esetén támogatást nyújtanak.

Egy többszintű-, hierarchikus címzési- és elnevezési sémában kellett gondolkodnunk, amivel megoldható, hogy a későbbiekben a hálózathoz esetleg újonnan hozzáadott felhasználók, illetve eszközök egyszerűen és könnyen felügyelhetők legyenek. Meg kellett oldani továbbá a külvilág felé megosztott tartalmakat (például a rögzített tárgyalási események) tároló eszközök egy, a belső hálózattól szigorúan szabályozott módon elkülönített, úgynevezett demilitarizált zónába (DMZ) történő elhelyezését. Ezzel a megoldással értük el azt, hogy a kívülről érkező támadások egy, a biztonsági szempontból érzékeny területtől jól elválasztott alhálózatba érkezzenek. A létrehozott belső címzési séma külvilágtól történő elrejtésére a Network Address Translations-t (NAT-ot), azaz a hálózati címfordítást alkalmaztuk. Segítségével a cég belső privát címei egy és ugyanazon publikus, mások által is látható címre történő lefordításával megakadályoztuk, hogy kívülről „felderíthető” legyen a hálózat struktúrája. A külső támadások mellett a belső forgalom szűrése is szükséges, így kiszűrhető az egyes hálózati területekre bejövő, illetve onnan származó nem megengedett csomagforgalom. Így a felesleges forgalom kiszűrésével a hálózat sávszélessége is javítható. Minden hálózati eszköz távolról történő elérését úgy kell biztosítani, hogy a megfelelő erősségű jelszavakkal védhetők legyenek a jogosulatlan hozzáférések elől.

A fenti szempontokat és tervezési megfontolásokat figyelembe véve, az alábbi szimulált hálózatot hoztuk létre.



Biztonságpolitikai követelmény volt, hogy távoli helyszínekről, ideértve a külső irodákat is, csak bizonyos helyi hálózati erőforrásokat legyenek elérhetőek. A szükséges szabályok a következők.

interface Serial0/0/0 (erre a portra állítjuk be) description Frame-Relay kapcsolat

ip address 172.18.0.18 255.255.255.252 encapsulation frame-relay (beágyazás)

frame-relay interface-dlci 200 (beállítás a kapcsolathoz) ip access-group 101 in (bejövő)

ip access-group 102 out (kimenő)

A távoli felhasználóknak (PC09, PC10) hozzá kell férni a P Serverhez, hogy láthassák a tartalmukat a weben keresztül,

access-list 101 permit tcp host 172.18.0.17 host 172.17.0.1 eq www

A távoli felhasználóknak képesnek kell lenni a P Serverekről fájlokat letölteni illetve oda feltölteni FTP segítségével.

access-list 101 permit tcp host 172.18.0.17 host 172.17.0.1 range 20 21 ftp

A távoli felhasználók használhatják a P Servert, hogy e-mailt küldjenek és fogadjanak SMTP és IMAP protokollok segítségével.

access-list 101 permit tcp host 172.18.0.17 host 172.17.0.1 eq smtp access-list 101 permit tcp host 172.18.0.17 host 172.17.0.1 eq 143

A távoli felhasználók nem érhetnek el semmilyen más szolgáltatást a P Serveren.
Megjegyzés: Az ACL-ek végén mindig van egy implicit tiltás, vagyis az engedélyezetten kívül minden tiltva van.

access-list 101 permit tcp any any implicit tiltás

Nem engedélyezett a központi iroda munkaállomásairól a távoli felhasználók munkaállomásai felé tartó forgalom. Minden olyan fájlt, amit a két helyszín között szükséges átvinni, a P Serveren kell tárolni, és onnan lehet őket FTP segítségével elérni (vissza irány engedélyezése).

access-list 102 permit tcp any any

Nem engedélyezett a távoli helyszín munkaállomásairól a központi helyszín munkaállomásai felé tartó forgalom.

access-list 102 deny ip 192.168.0.0 0.0.3.255 10.0.0.0 0.0.0.255

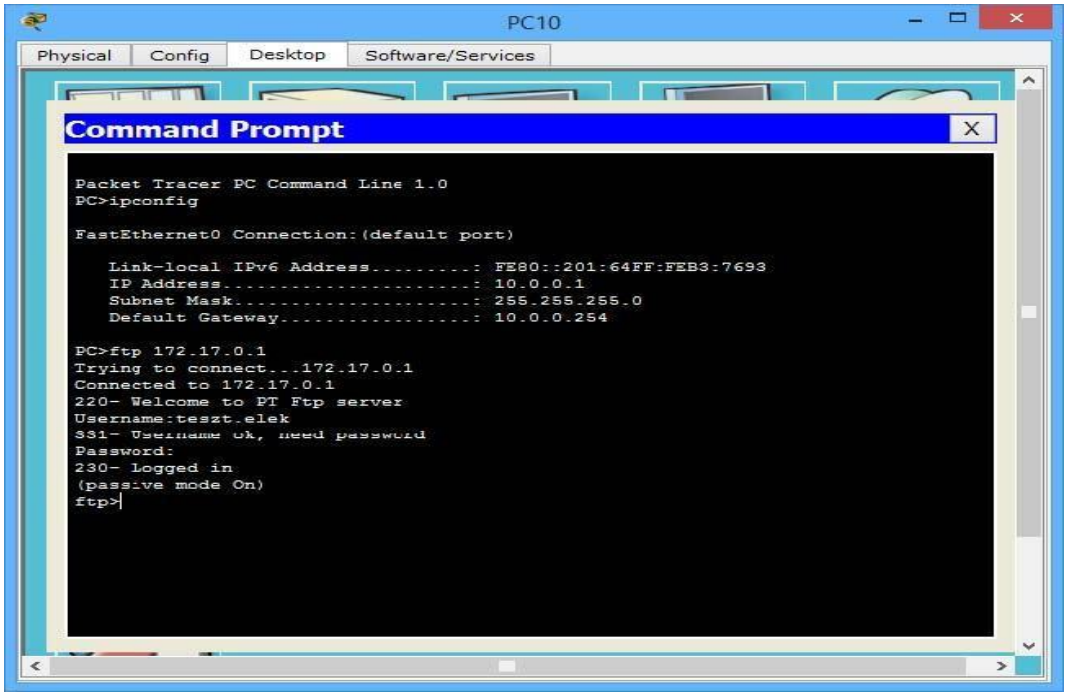
A központi irodában lévő routereket és switcheket csak a 70-es vlan munkatársai érhetik el Telnet segítségével.

line vty 0 4 access-class 70 in

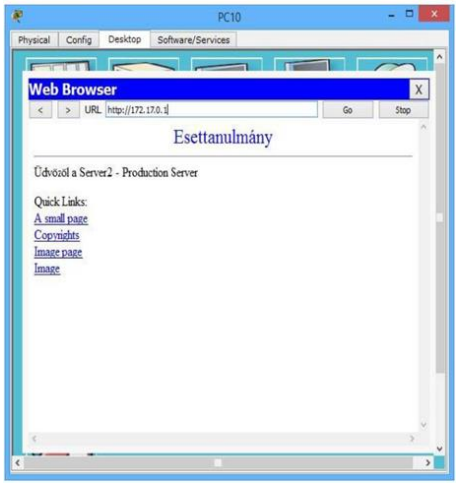
password 7 0822455D0A16 login

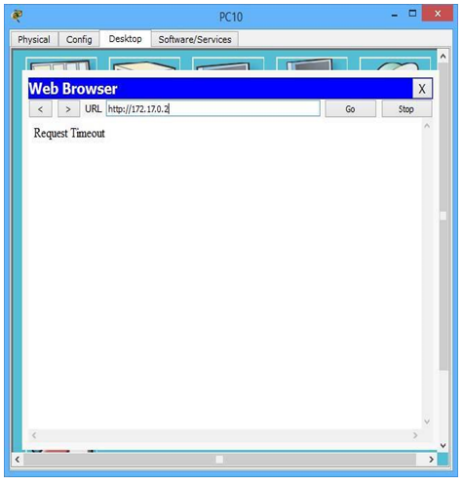
Az ACL beállítások tesztelése

Az alábbi tesztkimeneten látható, hogy a PC10-es távoli felhasználó az FTP protokollon keresztül éri el a P Servert. (2. ábra)



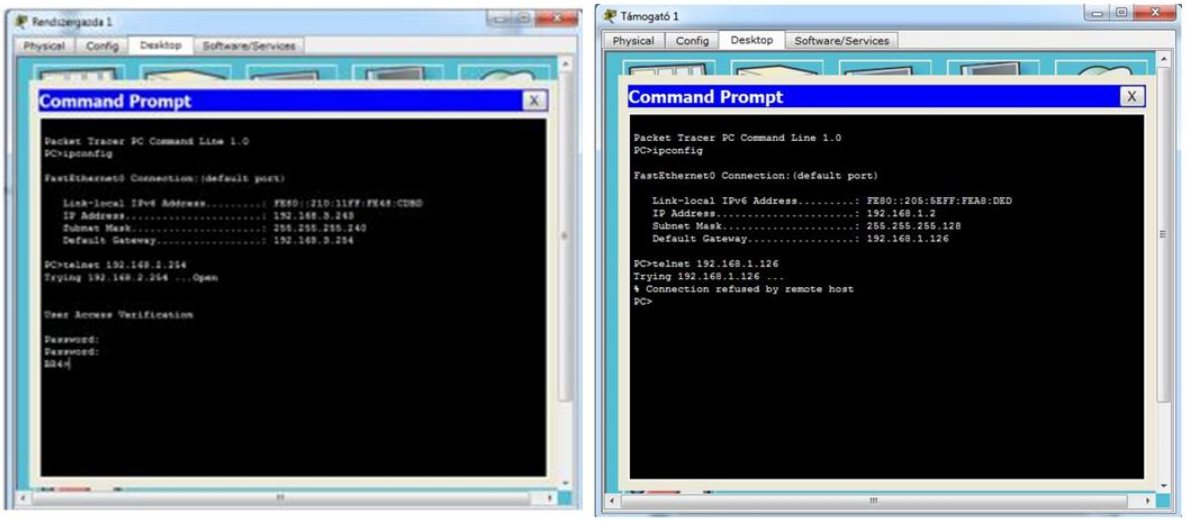
A következő két teszteredmény azt mutatja, hogy a távoli felhasználó HTTP protokollon keresztül éri el a P Server-t (3. ábra), illetve azt, hogy nem sikerül elérniük a Server 3-at (172.17.0.2).





A http teszt sikeres és sikertelen

A 70-es VLAN-ba (Pl: Rendszergazda 1) tartozó felhasználók elérik a hálózati eszközöket



Azon felhasználók, akik nem a 70-es VLAN-ba tartozó (Pl: Támogató 1) gépen dolgoznak, nem érik el telnet segítségével a hálózati eszközöket

Következtetések

A cikkben bemutattuk, hogy az ACL-kel nagyon egyszerűen és hatékonyan lehet szabályozni a hálózati forgalmat. Elsősorban a távoli használatra mutattunk be példákat, de természetesen ezt lehet bővíteni helyi hálózaton belül. Fontos, hogy a jól megtervezett, jól elhelyezett ACL-ek, nem csak biztonságosabb hálózatot biztosítanak, hanem elősegítik a hálózat jobb kihasználását, hatékonyabb működését is.

DHCP

DHCPv4 módjai

Manuális kiosztás

A rendszergazda egy előre meghatározott IPv4-címet rendel a klienshez, és a DHCPv4 csak átadja ezt az IPv4-címet az eszköznek.

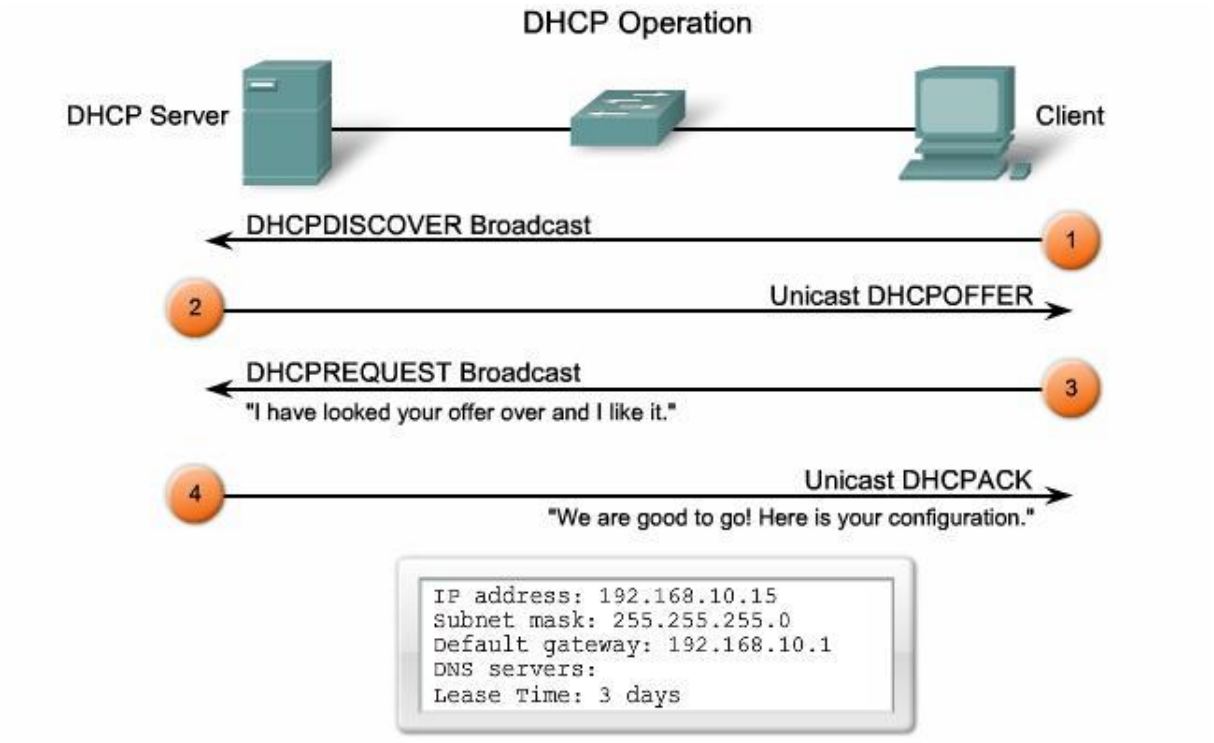
Automatikus kiosztás

A DHCPv4 automatikusan és maradandóan rendel hozzá egy statikus IPv4-címet az eszközhöz, amelyet a rendelkezésre álló címkészletből választ ki.

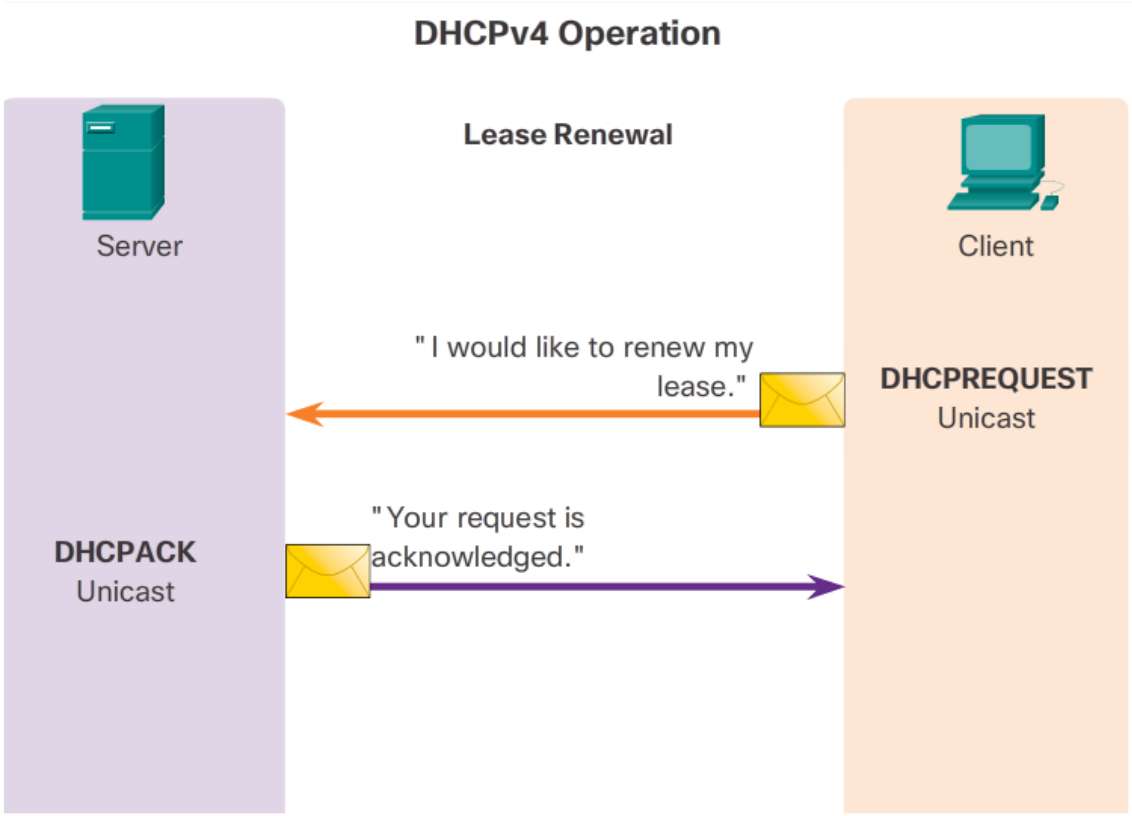
Nincs szó bérletről, a számítógéphez mindig ez a cím lesz hozzárendelve.

Dinamikus kiosztás

A rendelkezésre álló címkészletből a DHCPv4 dinamikusan oszt ki vagy ad bérbe egy IPv4-címet a szerver által meghatározott ideig (bérleti idő) vagy addig, ameddig a kliensnek szüksége van a címre.



DHCP működése

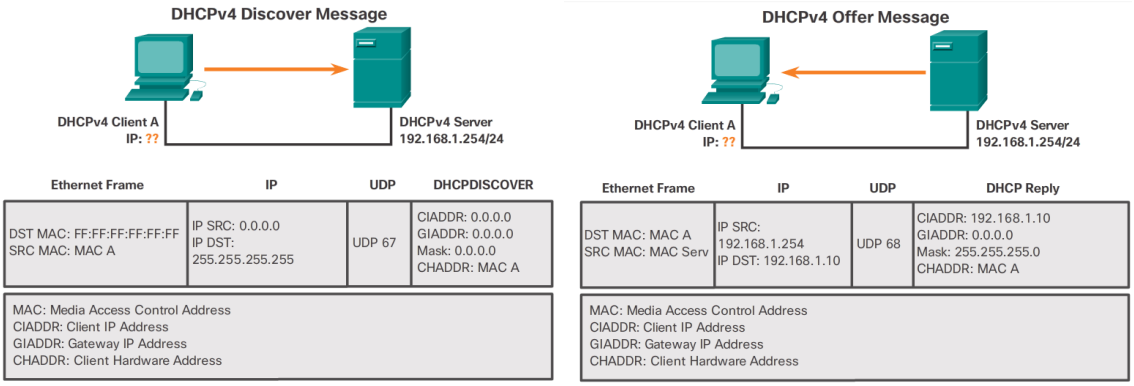


DHCP megújítása

- A DHCPv4-üzenetek beágyazása az UDP szállítási protokollon belül történik.
- A kliens által küldött DHCPv4-üzenetek az UDP 68-as forrásportját, illetve 67-es célportját használják.
- A szerver által a kliensnek küldött DHCPv4-üzenetek pedig az UDP 67-es forrásportját, illetve 68-as célportját használják.

8	16	24	32
OP Code (1)	Hardware Type (1)	Hardware Address Length (1)	Hops (1)
Transaction Identifier			
Seconds - 2 bytes		Flags - 2 bytes	
Client IP Address (CIADDR) - 4 bytes			
Your IP Address (YIADDR) - 4 bytes			
Server IP Address (SIADDR) - 4 bytes			
Gateway IP Address (GIADDR) - 4 bytes			
Client Hardware Address (CHADDR) - 16 bytes			
Server Name (SNAME) - 64 bytes			
Boot Filename - 128 bytes			
DHCP Options - variable			

DHCP üzenet formátuma



DHCP szerver beállítások egy Cisco-s eszközön

1. lépés: IPv4-címek kizárása
    - Az ip dhcp excluded-address paranccsal.
2. lépés: DHCPv4-készlet beállítása
    - Az ip dhcp pool készlet-neve paranccsal
3. lépés: Konkrét feladatok beállítása
    - network hálózat_cím alh._maszk
    - default-router ip_cím
    - dns-server ip_cím
    - domain-name tartománynév
    - netbios-name-server ip_cím

DHCP hibaelhárítás

1. hibaelhárítási feladat: Az IPv4-címek ütközésének feloldása
    - show ip dhcp conflict
2. hibaelhárítási feladat: A fizikai kapcsolat ellenőrzése
    - show interface interfész
3. hibaelhárítási feladat: A kapcsolat statikus IP-címmel történő ellenőrzése
4. hibaelhárítási feladat: A kapcsolóportok beállításainak ellenőrzése
5. hibaelhárítási feladat: A DHCPv4 egy alhálózaton vagy VLAN-on belüli működésének ellenőrzése.

Ha a kliens nem ugyanazon a hálózaton van , mint a szerver.

A forgalomirányító beállításainak ellenőrzéséhez végezzük el az alábbi lépéseket:

1. lépés Ellenőrizzük, hogy az ip helper-address parancs kadása a megfelelő interfészen megtörtént.
    - show running-config paranccsal.
    - show ip interface paranccsal.
2. lépés Ellenőrizzük, hogy a no service dhcp globális konfigurációs parancs nem lett-e kiadva.
    - Ez a parancs letilt minden DHCP-szerver és -közvetítő funkciót a forgalomirányítón.
    - show running-config paranccsal

- A DHCPv4-szerverként beállított forgalomirányítókon a DHCPv4-folyamat sikertelen, ha a forgalomirányítóhoz nem érkeznek DHCP-kérések a klienstől.
- A hibaelhárítás részeként ellenőrizzük, hogy a forgalomirányítóhoz valóban érkeznek-e DHCPv4-kérések.
- Ez a lépés egy ACL beállítását is magában foglalja, hibakeresés céljából.
- A kiterjesztett ACL a debug ip packet paranccsal együtt használva kizárólag a DHCPv4-üzeneteket jeleníti meg.
- Egy másik hasznos utasítás a DHCPv4 működésével kapcsolatos ibakereséshez a debug ip dhcp server events parancs.
- A címkiosztásokhoz és adatbázis-frissítésekhez hasonló szervereseményekről ad információt.
- Ezen felül a DHCPv4 küldések és fogadások dekódolására is használatos.

IPv6 címek beszerzése

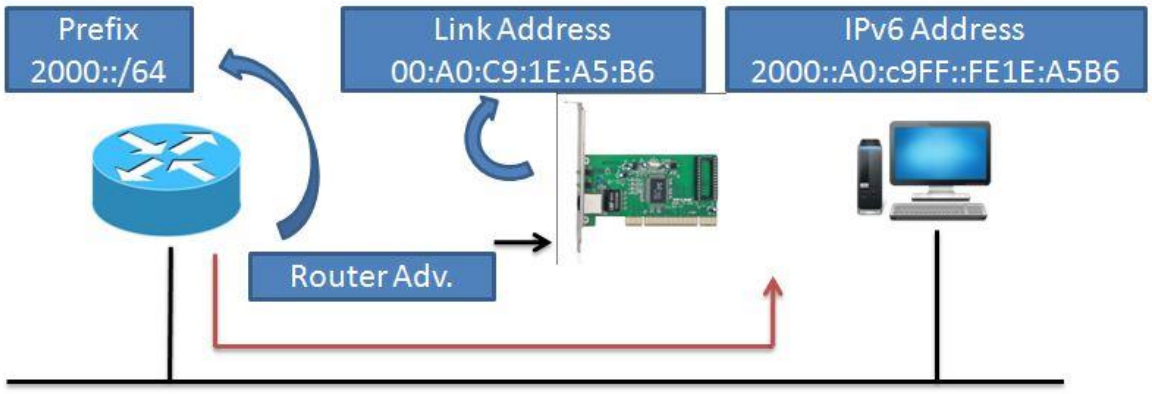
• Az IPv4-hez hasonlóan az IPv6 globális egyedi címek beállítása történhet manuálisan és dinamikusan egyaránt.
• Az IPv6 globális egyedi címek dinamikus kiosztására viszont két módszer is létezik:
– SLAAC (Stateless Address Autoconfiguration)
– Állapottartó DHCPv6

SLAAC (Stateless Address Autoconfiguration) Állapotmentes DHCPv6

• A SLAAC egy olyan módszer, amellyel egy DHCPv6-szerver szolgáltatásai nélkül szerezhetnek az eszközök IPv6 globális egyedi címet.
• A SLAAC működése az ICMPv6 protokollon alapul.
• Az ICMPv6 protokoll forgalomirányító-keresés (RS) és forgalomirányító-hirdetés (RA) üzeneteinek segítségével kínál címzési és egyéb konfigurációs adatokat, amelyeket normál esetben egy DHCP-szerver biztosítana.
• A SLAAC nem állapottartó.
• Ez azt jelenti, hogy nincs olyan szerver, amely karbantartja a hálózati címadatokat (használatban lévő, és melyek a rendelkezésre álló IPv6-címek).

Forgalomirányító-keresés (Router Solicitation, RS) üzenet
• Ha egy kliens úgy van beállítva, hogy a címzési adatokat automatikusan a SLAAC-tól kapja, akkor egy RS-üzenetet küld a forgalomirányítónak.
• Az RS-üzenetet az IPv6 összes forgalomirányítót magába foglaló (all-routers nevű) FF02::2 csoportcímére küldik el.
Az RA-üzeneteket a forgalomirányítók küldik, hogy címzési információt biztosítsanak azon klienseknek, amelyeket az IPv6-címük automatikus megszerzésére állítottak be.
• Az RA-üzenet tartalmazza a helyi szegmens előtagját és az előtag hosszát.
• A kliens ezen információk segítségével hozza létre a saját IPv6 globális egyedi címét.
A forgalomirányító rendszeres időközönként,
vagy egy RS-üzenetre válaszolva küld ki RAüzeneteket.

• Alapértelmezés szerint a Cisco forgalomirányítók 200 másodpercenként küldenek RA-üzenetet.
• Az RA-üzeneteket mindig az IPv6 összes állomást tartalmazó (all-nodes nevű) FF02::1 csoportcímére küldik.



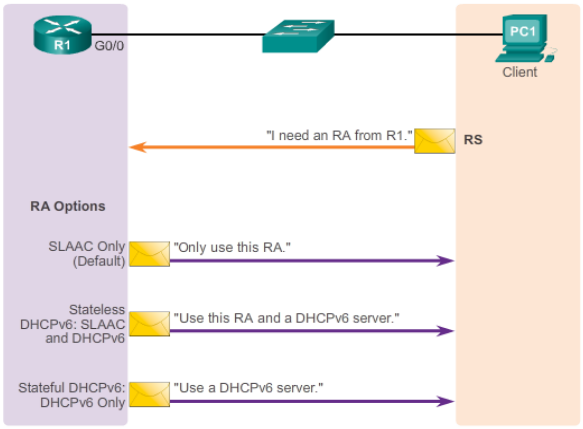
icmpv6 stateless address autoconfiguration

Kliens, hogyan generál IPv6 címet?

• Annak, hogy a PC1 létrehozhassa a saját, egyedi IID-jét, két módja van:
– EUI-64 - Az EUI-64 folyamat segítségével a PC1 a 48-bites MAC-címe alapján hoz létre egy IID-t.
– Véletlenszerűen generált - A 64-bites IID lehet egy, a kliens operációs rendszere által generált véletlen szám.

SLAAC és DHCPv6
• Az, hogy egy kliens az IPv6-címadatait a SLAAC vagy a DHCPv6 használatával, esetleg a kettő kombinációjával szerezze meg, az RA-üzenetben lévő beállításokon múlik.
– Az ICMPv6 RA-üzenetei két jelzőbitet tartalmaznak, annak jelölésére, hogy a kliensnek melyik lehetőséget kellene választania.
– Az egyik a felügyelt címkonfigurációs jelzőbit (Managed Address Configuration flag, M jelzőbit),
– a másik az egyéb konfigurációs jelzőbit (Other Configuration flag, O jelzőbit).

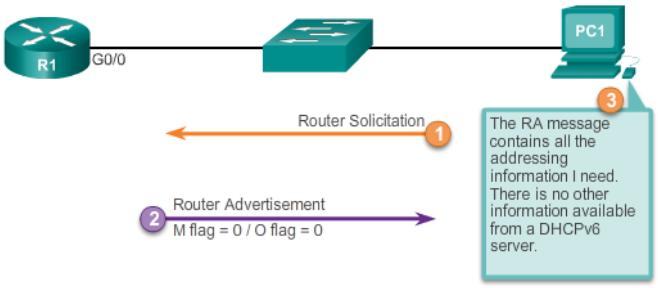
M és O jelzőbit
• Az M és az O jelzőbit különböző kombinációjával, az RA-üzenetek három címzési lehetőség egyikét kínálják az IPv6 eszköz számára:
– SLAAC (csak forgalomirányító-hirdetéssel)
– Állapotmentes DHCPv6 (forgalomirányítóhirdetéssel és DHCPv6-tal)
– Állapottartó DHCPv6 (csak DHCPv6-tal)



SLAAC és DHCPv6

SLAAC opció (csak forgalomirányítóhirdetéssel)

• A Cisco forgalomirányítókon a SLAAC az alapértelmezett lehetőség.
– Ilyenkor az M és az O jelzőbit értéke egyaránt 0 az RAüzenetben.
• Ez a lehetőség arra utasítja a klienst, hogy kizárólag az RA-üzenet információit használja.
• Ebbe beletartoznak az előtagra, az előtag hosszára, a DNS-szerverre, az MTU-ra, valamint az alapértelmezett átjáróra vonatkozó adatok.



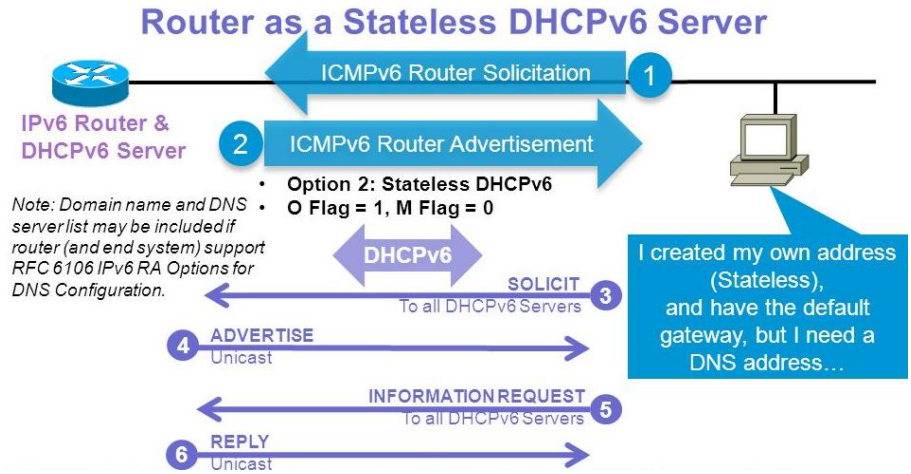
SLAAC option

• Az RA-üzenetek beállítása a forgalomirányítók egyes interfészein történik.
• A SLAAC működésének újbóli engedélyezése
– Ezt az alábbi, globális konfigurációs parancsokkal tehetjük meg:
– **`Router(config-if)# no ipv6 nd managed-config-flag`**
– `Router(config-if)# no ipv6 nd other-config-flag`

Az állapotmentes DHCPv6 opció

(forgalomirányító-hirdetéssel és DHCPv6-tal)
• Az állapotmentes DHCPv6 opció értesíti a klienst, hogy az RA-üzenet információit kell használnia a címzéshez, de a többi konfigurációs beállítást a DHCPv6-szerver bocsátja a rendelkezésére.
– Az RA-üzenetben szereplő előtaggal és az előtag hosszával a kliens létrehozza a saját IPv6 globális egyedi címét.
– A kliens ezután kommunikálni kezd egy DHCPv6-szerverrel, hogy megszerezze az RA-üzenetben nem szereplő további adatokat.

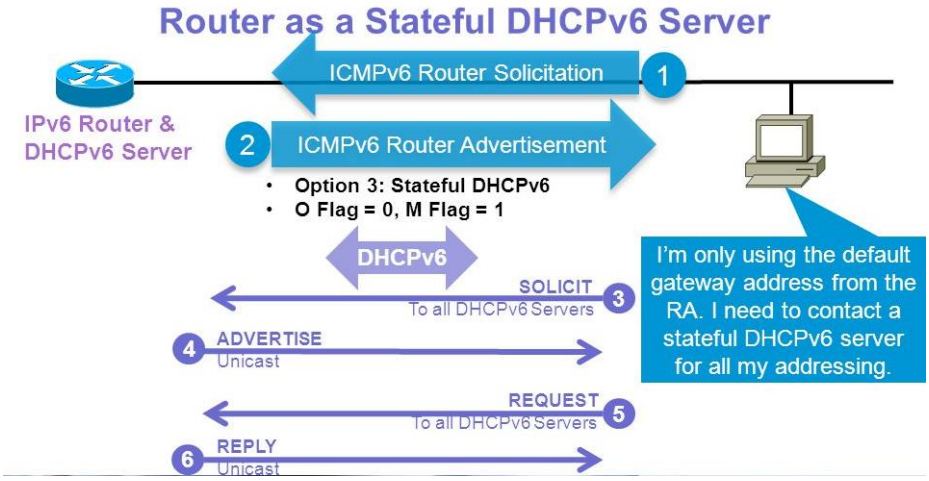
– Ezt a folyamatot állapotmentes DHCPv6-nak nevezzük, mivel a szerver semmilyen, a kliens állapotára vonatkozó információt nem tart karban.



• Az állapotmentes DHCPv6 esetében az O jelzőbit értékét 1-re kell állítani, az M jelzőbit értéke pedig maradjon az alapértelmezés szerinti 0.
• Ahhoz, hogy egy forgalomirányító interfészén kiküldött RA-üzenetet úgy módosítsunk, hogy az állapotmentes DHCPv6-ot jelöljön, használjuk az alábbi parancsot:
– **`Router(config-if)# ipv6 nd other-config-flag`**

Állapottartó DHCPv6 (csak DHCPv6-tal)

• Ez a lehetőség hasonlít legjobban a DHCPv4-hez.
• Ebben az esetben az RA-üzenet azt jelzi a kliensnek, hogy ne használja az RA-üzenet információit.
• Így minden címzési és konfigurációs adatot egy DHCPv6-szervertől kell megszereznie.
• Ezt állapottartó DHCPv6-nak nevezzük, mivel a DHCPv6-szerver tartja karban az IPv6-os állapotinformációkat.



• Az M jelzőbit mutatja, hogy szükség van-e az állapottartó DHCPv6 használatára. Az O jelzőbitnek ebben nincs szerepe.
• Az állapottartó DHCPv6 kifejezhető azzal, ha az alábbi parancs segítségével átállítjuk az M jelzőbit értékét 0-ról 1-re:

– Router(config-if)# ipv6 nd managed-config-flag
• A DHCPv6-üzenetek küldése UDP-n keresztül történik.
• A szerver által a kliensnek küldött DHCPv6-üzenetek az UDP 546-os célportját használják.
• A kliens által a szervernek küldött DHCPv6-üzenetek pedig az UDP 547-es célportját használják.

• A DHCPv6 SOLICIT üzenet

– all-DHCPv6-servers FF02::1:2 csoportcímre.
– Ez a csoportcím link-local hatókörrel rendelkezik,
=> a forgalomirányítók nem továbbítják az üzeneteket más hálózatokba.
• DHCPv6 REQUEST vagy INFORMATIONREQUEST
üzenet
– Állapotmentes DHCPv6-kliens esetén
INFORMATION-REQUEST üzenetet küld
• Kizárólag a konfigurációs beállításokat kéri.
– Állapottartó DHCPv6-kliens esetén DHCPv6 REQUEST üzenetet küld
• Az IPv6 címet és az összes egyéb konfigurációs beállítással együtt a szervertől szerezze meg.

Állapotmentes DHCPv6 szerver konfigurálás

• 1. lépés: Az IPv6-irányítás engedélyezése
– Az ipv6 unicast-routing paranccsal engedélyezhető az IPv6-irányítás. (ICMPv6 RA-üzeneteinek küldéséhez).
• 2. lépés: DHCPv6 készlet beállítása
– Az ipv6 dhcp pool készlet-neve paranccsal.
• 3. lépés: A készlet paramétereinek beállítása
– Pl: DNS-szerver címe, valamint a tartomány neve.
• 4. lépés: A DHCPv6-interfész beállítása
– ipv6 dhcp server készlet-neve
– ipv6 nd other-config-flag

Ha a kliens egy router

• Az interfész link-local címe automatikusan létrejön, ha az interfészen engedélyezett az IPv6-protokoll.
– Ehhez az ipv6 enable parancsot használjuk.
– Miután a forgalomirányító megkapja a link-local címét, küldhet RS-üzeneteket és részt vehet a DHCPv6-folyamatban is.
• Az ipv6 address autoconfig paranccsal engedélyezhető az IPv6-címadatok SLAAC használatával történő automatikus beállítása.

Állapottartó DHCPv6-szerver

• 1. lépés: Az IPv6-irányítás engedélyezése

– ipv6 unicast-routing paranccsal
• 2. lépés: DHCPv6 készlet beállítása
– Az ipv6 dhcp pool készlet-neve paranccsal
• 3. lépés: A készlet paramétereinek beállítása
– address prefix/hossz
– lifetime sec
– a DNS-szerver címe,
– a tartomány neve.
• 4. lépés: Az interfészkonfigurációs parancsok
– ipv6 dhcp server készlet-neve interfészkonfigurációs parancs
– Az M jelzőbit értékét át kell állítani 0-ról 1-re, az ipv6 nd managedconfig-flag paranccsal.

Állapottartó DHCPv6 kliens

• az ipv6 enable interfészkonfigurációs paranccsal, engedélyezzük hogy a forgalomirányító megkapja a link-local címét, küldhessen RS-üzeneteket és részt vehessen a DHCPv6-folyamatban is.
• Az ipv6 address dhcp interfészkonfigurációs paranccsal engedélyezhetjük, hogy a forgalomirányító DHCPv6-kliensként viselkedjen ezen az interfészen.

Ha a DHCPv6-szerver és a kliens eltérő hálózaton vannak

• A DHCPv6-közvetítő beállítása
– Az ipv6 dhcp relay destination paranccsal.
• Ezt a parancsot a DHCPv6-kliens felőli interfészen kell kiadni, célként pedig a DHCPv6-szerver címe legyen megadva.

DHCPv6 hibaelhárítási feladatok

• 1. hibaelhárítási feladat: Címütközések feloldása
– A show ipv6 dhcp conflict parancs megjeleníti az állapottartó DHCPv6-szerver által naplózott címütközéseket.
• 2. hibaelhárítási feladat: A címkiosztási mód ellenőrzése
– A show ipv6 interface interfész parancs használatával
• 3. hibaelhárítási feladat: A kapcsolat statikus IP-címmel történő ellenőrzése
• 4. hibaelhárítási feladat: A kapcsolóportok beállításainak ellenőrzése
• 5. hibaelhárítási feladat: A DHCPv6 egy alhálózaton vagy VLAN-on belüli működésének ellenőrzése

NAT

Címkezelés problematikája

- Az Internetes hálózatokban ahhoz, hogy elérhetővé váljanak az egyes hálózatok egymás számára, globálisan érvényes címeket használunk.
- Hogy csökkentsük a felhasznált címek számát és növeljük a biztonságot, autonóm hálózatainkon (Intraneteinken) belül lokálisan érvényes címeket használunk. Ezek érvénytelenek az Interneten.
- Ezért autonóm rendszereink határán a lokális címeinket globális címre, vagy címekre fordítjuk. Ezt a megoldást hívjuk címfordításnak. (NAT/PAT)

NAT - Network Address Translation

A NAT egy címtranszformációs eljárás, melyben IP címeket képezünk le egy címzési övezetből egy másikba

Tulajdonságok:

- Transzparens címösszerendelés (nyilvántartás alapján)
- Transzparens forgalomirányítás biztosítása
- ICMP hibaüzenetek adatrészének transzformációja

Megvalósítása:

- Statikus összerendelés
- Dinamikus összerendelés

Előnyei:

- Privát címek használata a belső hálózaton (címtakarékosság)
- Biztonság növelése (a belső hálózat cím-struktúrája nem látható a külső hálózatról)

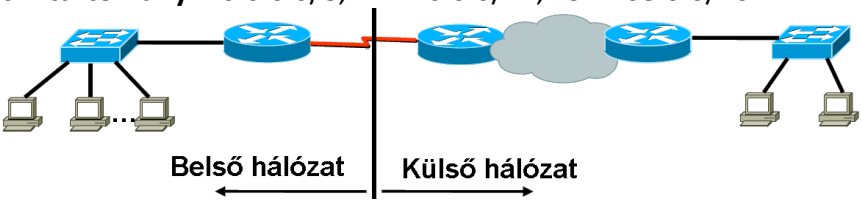
Hátrányai:

- A címfordítást végző eszközre jelentős terhelést ró
- Azon protokollok, amelyek a csomagok adatrészében cím információkat továbbítanak, csak akkor NAT tűrőek, ha a címfordítást végző eszköz képes az adatrészben továbbított tartalomban is elvégezni a szükséges címek cseréjét! (Application Level Gateway – ALG funkció)

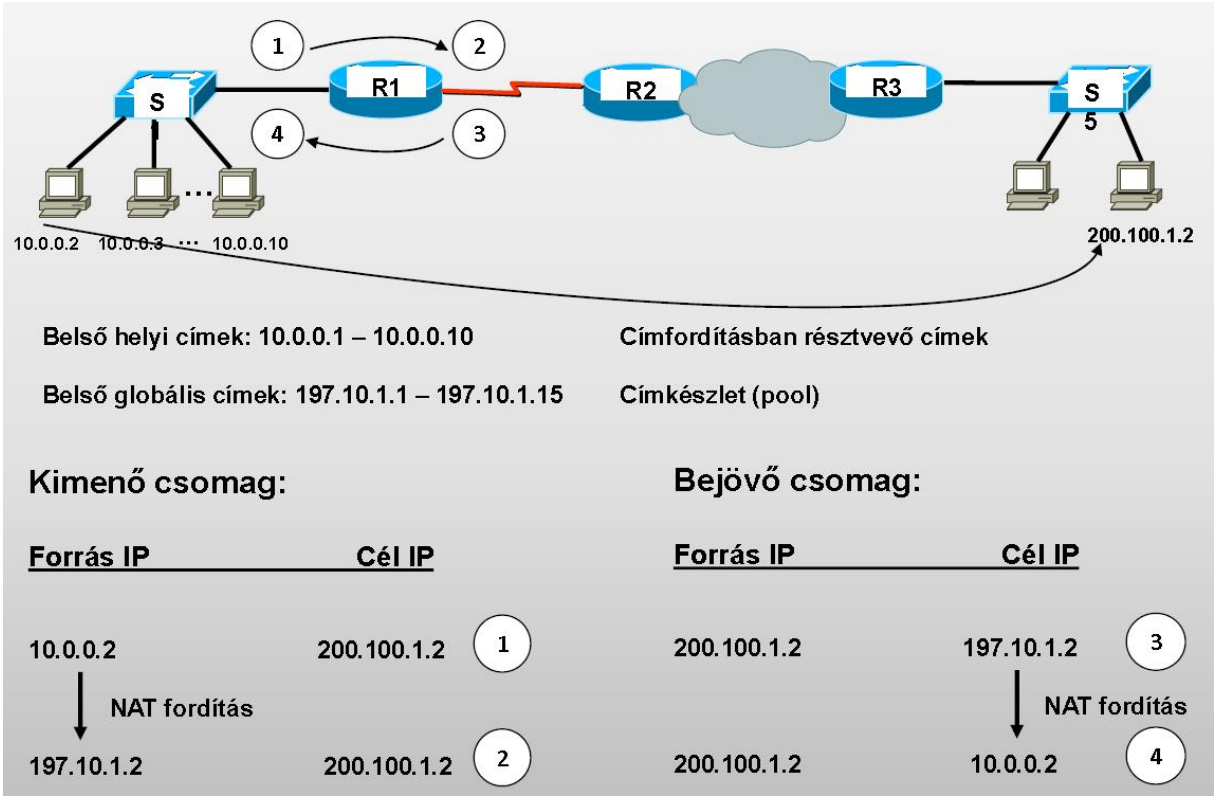
Fogalmak

- **Címzési övezet** (address realm): Az a hálózatrész, amelyben biztosítani kell az IP címek egyediségét
- **Külső hálózat** (public/global/external network): Az IANA által kezelt címtartománnyal rendelkező címzési övezet

- **Belső hálózat** (Private/Local Network): Az intézmény saját (belső) címzéssel rendelkező címzési övezete.
- **Privát címtartomány**: 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16



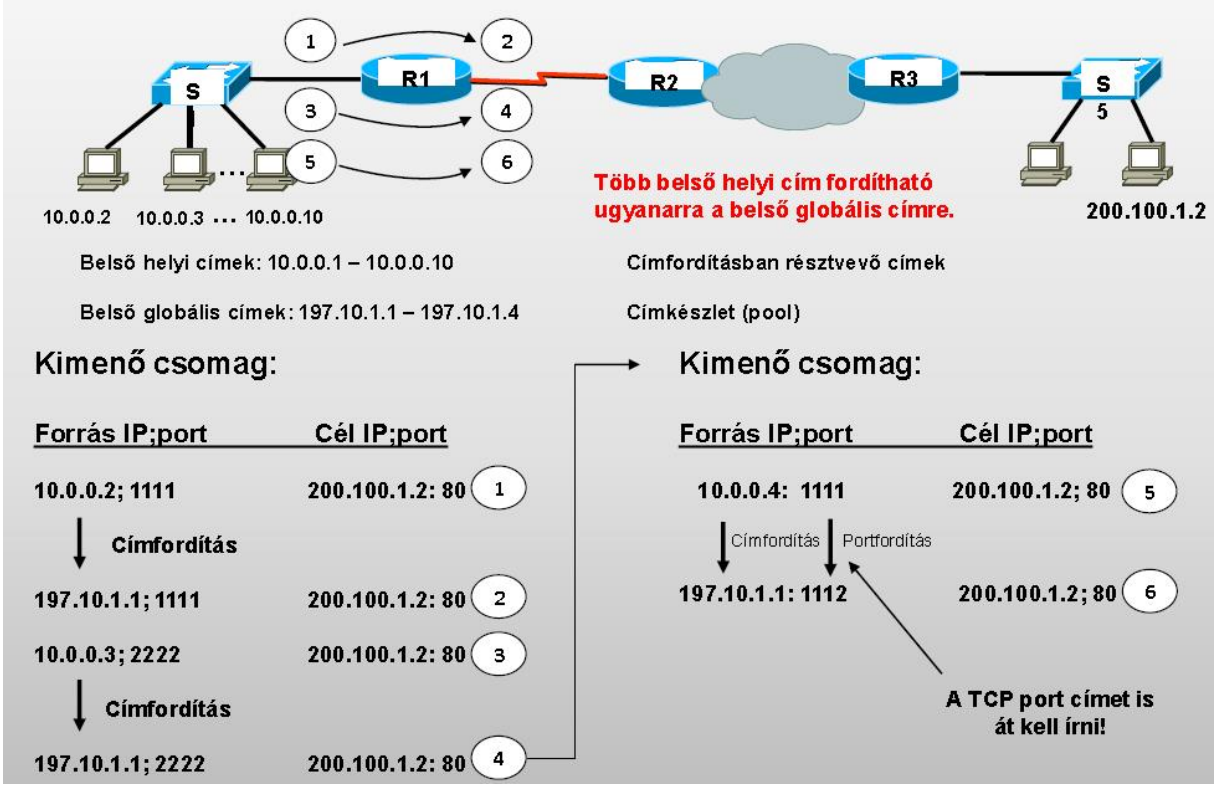
- **Lokális cím:** A belső hálózaton használt cím
- **Globális cím:** A külső hálózaton használt cím
- **Belső lokális cím**: Egy belső hálózati csomópont IP címe a belső hálózaton
- **Belső globális cím**: Egy belső hálózati csomópont IP címe a külső hálózaton
- **Külső lokális cím**: Egy külső hálózati csomópont IP címe a belső hálózaton
- **Külső globális cím:** Egy külső hálózati csomópont IP címe a külső hálózaton



Példa a NAT-ra

PAT - Port Address Translation

A NAT olyan megvalósítása, amikor lehetséges, hogy a címfordítás során a belső lokális és a belső globális címek között nem 1:1 típusú megfeleltetést alkalmazunk, hanem egy-egy globális IP-címnek több lokális címet is megfeleltetünk: N:1 típusú leképzés
A transzformációkat tartalmazó táblázatot ki kell egészíteni a Belső lokális és globális címeken túl a TCP/UDP port vagy az ICMP saquece number értékével.



PAT példa

Párosítások

Belső lokális cím	Belső globális cím	Mód
1 db	1 db	NAT
Több	1 db	PAT
Több	Több (pool)	NAT/PAT
1 db	Több (pool)	NAT

Dinamikus társítás: A kommunikáció csak a belső hálózatról a külső hálózat irányába kezdődhet. Majd a dinamikusan létrejött bejegyzés útján a válaszcsomagok visszaérkezhetnek. De a külső hálózat nem kezdeményezheti a kommunikációt!

Statikus társítás: Adott eszközt mindig ugyan arra a címre képez le. Emiatt a külső hálózatból is kezdeményezhető a kommunikáció. PAT esetén Port Forward alkalmazható.

Dinamikus PAT konfiguráció

Minden belső lokális cím az egyetlen belső globális címet használja ami nem más, mint a külső interfész címe is egyben!

Címfordításban résztvevő gépek megadása:
```
Router(config)# access-list 1 permit 10.0.0.0 0.0.0.255
```

A címfordítás interfészének megadása:
```
Router(config)# ip nat inside source list 1 interface Serial
2/0 overload
```

A címfordítás belső és külső interfészei:
```
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface Serial 2/0
Router(config-if)# ip nat outside
Router(config-if)# exit
```

Statikus NAT konfiguráció

A tipikus 1-1 kapcsolat megvalósítása

Statikus címfordító bejegyzés:
```
Router(config)# ip nat inside source static 10.0.0.254
197.10.1.14
```

A címfordítás belső és külső interfészei:
```
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip nat inside
Router(config-if)# exit
Router(config)# interface Serial 2/0
Router(config-if)# ip nat outside
Router(config-if)# exit
```

Statikus PAT konfiguráció

A tipikus N-1 kapcsolat megvalósítása port továbbítással

Statikus cím és port fordító bejegyzés:
```
Router(config)# ip nat inside source static tcp 10.0.0.254 80
197.10.1.14 80
Router(config)# ip nat inside source static tcp 10.0.0.253 22
197.10.1.14 22
```

A címfordítás belső és külső interfészei:
```
Router(config)# interface FastEthernet 0/0
Router(config-if)# ip nat inside
```

```
Router(config-if)# exit
Router(config)# interface Serial 2/0
Router(config-if)# ip nat outside
Router(config-if)# exit
```

STP működése

A protokoll működése

- A switchek a portokon BPDU-kkal kommunikálnak
- A BPDU tartalmazza:
    - port információ (sebesség, prioritás)
    - switch információ (prioritás, MAC, időzítők
    - prioritások, vlan információk
- A protokoll először kitalálja, hol legyen a feszítőfa gyökere
- Ha ez megvan, az információt elterjeszti, hogy mindenki kiszámolhassa az
- optimális útvonalat oda
- Megtörténik a port szerepek kiosztása
- Közben megtörténnek a port állapot-változások
- A hálózat működőképes lesz, mert
    - A protokoll felépített egy feszítőfát, ami körmentes
    - A protokoll megtalálta a hurkokat és ezeket lekapcsolta
    - A protokoll képes arra, hogy topológiaváltás esetén az alternatív útvonalakat engedélyezze
    - Ezáltal minden időpontban pontosan egy útvonal lesz bármely két csomópont között

Az STP implementációk különbségei

- Az idő, ami a felfedezéshez szükséges
- Az idő, ami a konvergáláshoz szükséges
- A tartalék útvonalak előre kiszámolása
- A lehetséges port állapotok száma
- A szükséges erőforrások mennyisége

A választás menete

1) Bekapcsolás után minden switch root-nak hiszi magát
2) Teljesen mindegy milyen sorrendben, az egyik switch elkezd BPDU-kat küldözgetni, magát root-nak hirdetve
3) Amennyiben az ezt meghalló másik switch prioritása magasabb számérték, tehát kevésbé root, a hirdetést elfogadja és lemond a saját root szerepéről
4) Amennyiben az ezt meghalló másik switch prioritása alacsonyabb számérték, tehát jobban root, a hirdetés alapján megállapítja, hogy a hirdető tévesen hiszi magát root-nak
5) A következő lépésben mindazok a switchek, amelyek magukat a hallott hirdetéssel szemben jobb root-nak tartják magukat, ezt hirdetni kezdik ugyanúgy, ahogy a korábbi hirdetés történt

6) Azok a switchek akiknek nem osztottak lapot (már a 3. lépésben kiestek a választáson) némán figyelik a nagyok csatáját, mindig feljegyezve a legjobb hallott hirdetés feladóját, mint root switchet
7) Azok a switchek, akik még mindig root-nak hiszik magukat előbb utóbb meghallják a náluk jobb prioritással bíró hirdetéseket és lemondanak a root szerepről

A választás utáni lépések

1) Pontosan egy root switch van és mindenki tudja melyik switch ID volt az
2) Minden nem-root switch kiválaszt egy portot, amelyik a root felé néző port lesz (Root Path Cost alapján)
3) Minden szegmensben a legmagasabb RPC-jű nem-root port designated port lesz (szegmens a kábel a switchek között)
4) Ebből következik, hogy a root switchen csak designated port van (mert ő maga a root)
5) A root és designated portok elkezdenek tanulni, majd forgalmazni
6) Minden port ami se nem root, se nem designated, ún. non-designated port, blocking állapotba kerül
7) Ezen a ponton a hálózatban nincs kör, az STP konvergencia befejeződött
8) Ebben az állapotban a switchek a megfelelő portokon továbbra is BPDU-kat küldenek, a blocking port felé is
9) Amennyiben 20 másodpercig nem érkezik BPDU, a switch hibát feltételez és a blocking portot felengedi

Az STP jellemzői

- Menet közben folyamatos a kommunikáció minden porton
- A blocked port csak adatforgalmat nem engedélyez, BPDU-k jönnek-mennek rajta
- Bármilyen kábelhiba, szakadás valahol BPDU hiányt idéz elő
- A BPDU hiány 20 mp. múlva STP konvergenciát indít el
- Ekkor a blokkolt portok felengedése kezdődik el, tanulási folyamat mellett
- A teljes hálózati kiesés a felhasználóknak kb. 50 mp, ez a BPDU timer (20 sec) és a tanulási idő (2x15 sec)
- Üzleti környezetben ez nem tolerálható (= rapid STP, amely gyorsabban konvergál)
- A klasszikus STP nem vette figyelembe a VLANokat (látható, hogy a vlan 1 és vlan 3 topológiája megegyezik)
- A PVST, PVST+ minden VLAN-ra külön-külön futtat STP-t amik különbözőek lehetnek
- Sok VLAN esetén (sok száz, néhány ezer) ez rendkívül erőforrás-igényes
- Ez vezetett az MST-hez (nem VLAN-onként, hanem VLAN csoportonként van STP)

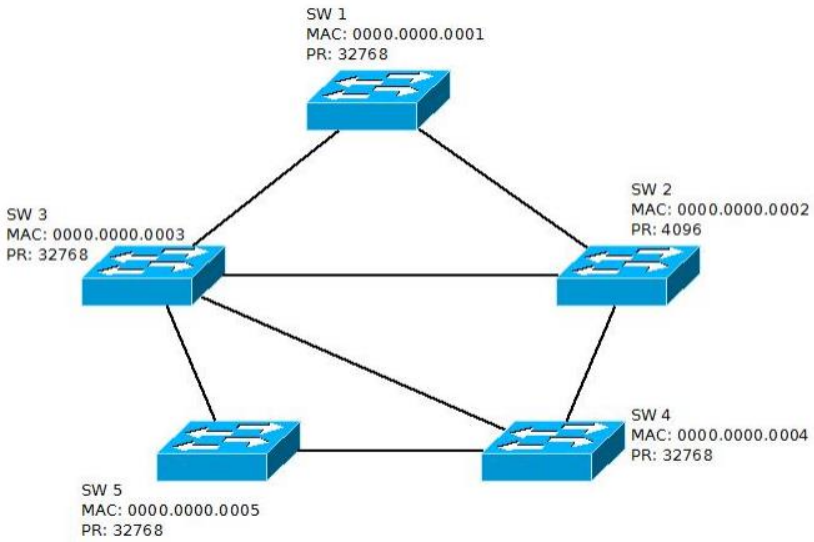
Előnyök:

- Jelentkezzen, aki hibától számítva kb. 30-50 mp alatt azonosít és megjavít egy kábelezési hibát
- Dinamikus alkalmazkodás automatikusan a mindenkori legjobb helyzethez (sebesség, stb)
- Mostantól bármikor ha egy kábelt ki akarsz húzni, bátran, legfeljebb kb. 50 mp után helyreáll minden
- Nem kell a javításokkal, átkábelezésekkel többet munkaidő utánra várni, csökkenhet a túlóra

Hátrányok:

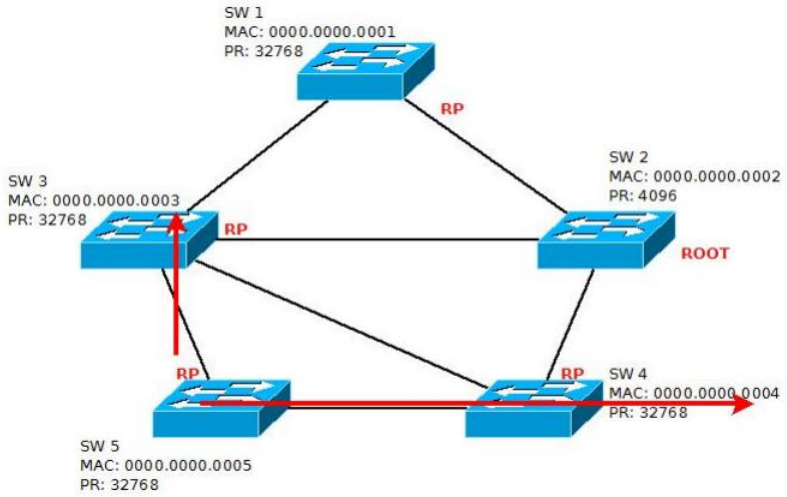
- Nem triviális protokoll, komplex hibák forrása lehet
- A 30 másodperc kiesés is sok lehet, a konvergencia lassú
- Tervezést, odafigyelést igényel a bevezetése, elkerülendő a nemkívánt root switch válaszási eredményeket
- Minden switchport minden alkalommal az első 30 másodpercben tanul, nem lehet forgalmazni (windows boot)
- Együttműködés más gyártók switcheivel problémás lehet
- Minden eszköznek támogatnia kell, hogy valóban hurokmentes legyen a topológia
- Az extra kábelek és extra portok extra pénzbe kerülnek és ezek az idő nagy részében kihasználatlanok

Az STP szemléltetése



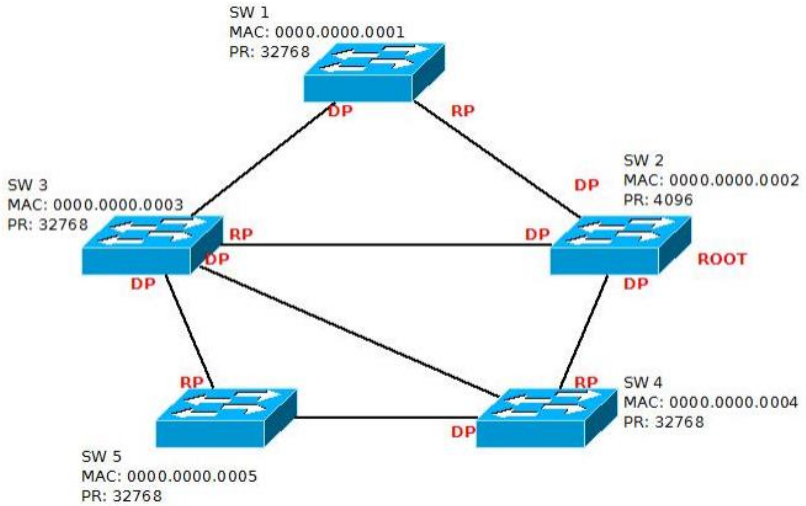
- Kiinduló állapot, mindenki root-nak hiszi magát
- Megkezdődnek a hirdetések (prioritások + MAC címek)
- Látható, hogy SW2 lesz a root, mivel a prioritása neki a legkisebb
- Erre legkésőbb SW5 fog rájönni, mivel ő két ugrásnyira van SW2-től
- A prioritás értéke miatt SW2 győzött
- A következő lépés a root portok (RP) meghatározása

- Minden switch, aki direktben kapcsolatban áll SW2-vel, automatikusan RP-nek jelöli a portot, ami SW2 felé néz
- Egyedül SW5 nem áll közvetlen kapcsolatban SW2-vel, itt nem triviális, melyik port



lesz RP

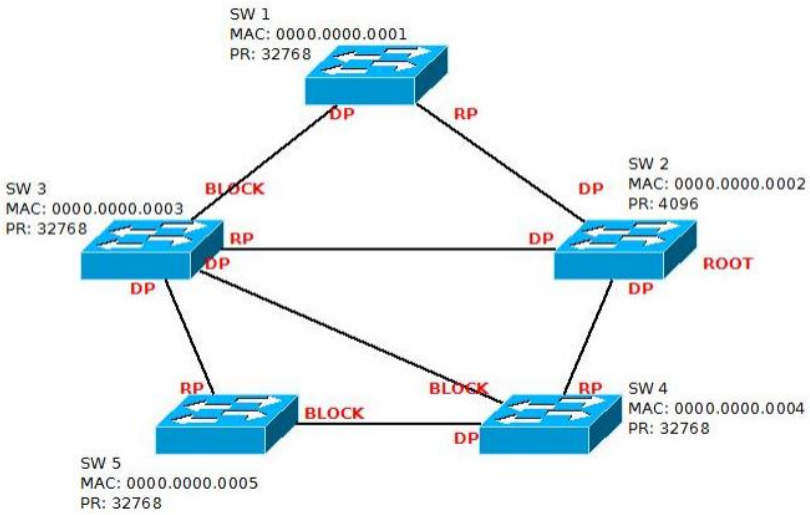
- Az ábrán bejelölve láthatók a root portok
- SW5 a kisebb RPC (Root Path Cost) felé fog RP-t választani
- Mivel ebben a hálózatban minden port egyenlő sebességű (mondjuk 100 mbit/s), a költség mindenhol 19
- Az egyetlen különbség a switchek MAC címe: SW3 jobb, mint SW4, így az RP SW3 felé fog mutatni
- Amennyiben mondjuk SW3 és SW5 között 10 mbit/s lenne a kapcsolat, nyilván SW4 felé lenne az RP
- A következő lépés minden RP-vel szemben egy DP-t jelölni, hiszen az RP-vel szemközti oldal DP lesz



- Minden RP-vel szembeni DP-t megjelöltünk
- Hátra vannak a szerepek kijelölése SW1 és SW3, SW3 és SW4, valamint SW4 és SW5 között

- A fenti három kábel felesleges, mivel mindkét végük olyan switchbe van kötve, aminek már van RP-je
- Az RP egy jobb (gyorsabb, közvetlenebb) kapcsolatot jelöl, tehát ezek a kábelek nem lehetnek jobbak – ilyen volt a kiválasztási algoritmus
- Ezek a kábelek tehát feleslegesek, ezeken
- valahol blokkolni kell, mert kört okoznak

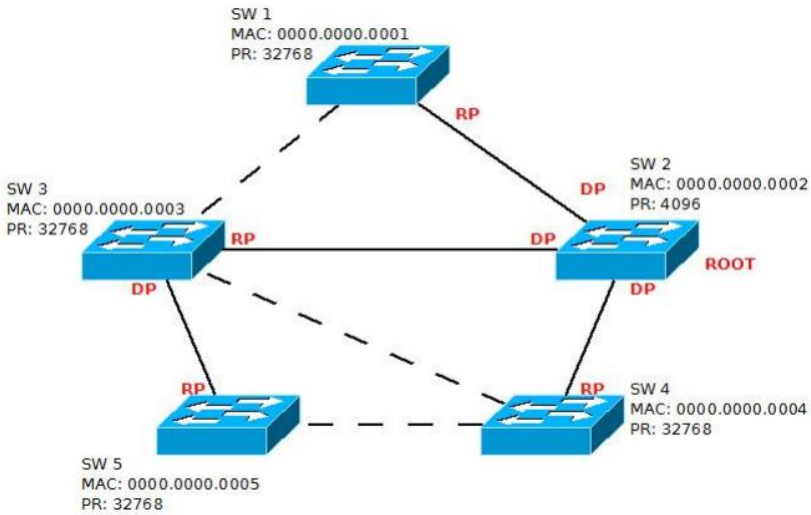
- A kábelek “jobb” felén tehát DP lesz, a rosszabb felén pedig blokkolás



- A három kábelen (SW1 – SW3, SW3 – SW4, SW4 – SW5) bejelölve a “jobbik” oldal, ahol DP lesz

- A velük szemközti portok lesznek blokkolva

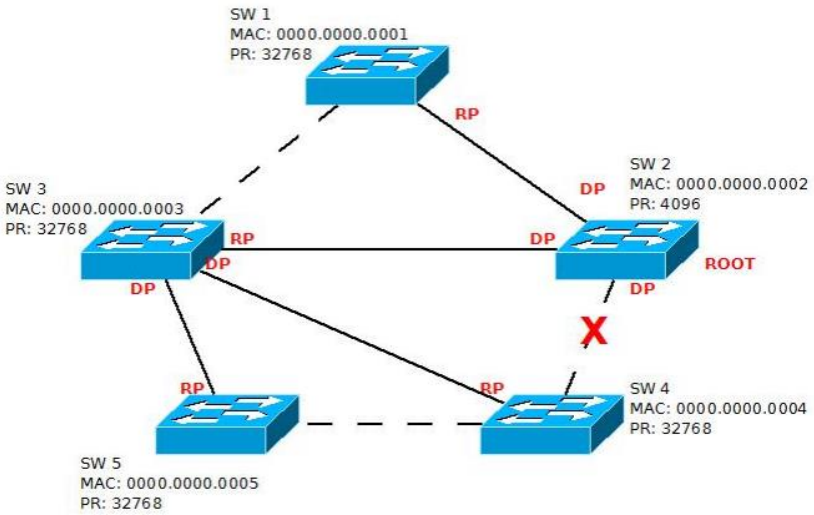
- Megjelölve minden port, a blokkolt portok is
- Ezzel a topológiát a switchek feltérképezték a küldött BPDU-k segítségével
- Összeállt a végleges feszítőfa, amely már nem tartalmaz kört



- A szaggatottal jelölt kábeleket a switchek egy oldalon blokkolták, szakadást okozva

- A folyamatos vonalon zajlik a hálózati forgalom, az STP konvergencia végetért
- Pontosan látható, hogy a root switch választás kritikus fontosságú
- Ha nem állítjuk be kézzel a prioritásokat, a MAC címek győznek
- Előfordulhat, hogy egy csoffadt, régi switch nyeri a root címet, rajta megy át minden forgalom
- Ezzel esetleg meg sem tud küzdeni: lefagy, újraindul, instabil lesz
- A root switch kritikus fontosságú, ha kiesik, átmenetileg izolált szigetekre szakad a hálózat

Nézzük meg, mi történik kábelhiba esetén?



- A piros X jelzi a kábelhiba helyét, itt eredetileg ment forgalom
- A szakadás pillanatától kezdve SW4 nem kap BPDU-kat SW2-től
- Viszont továbbra is kap BPDU-kat SW3-tól és SW5-től
- A dead timer lejárta után (20 mp) SW4 szakadtnak nyilvánítja a hibás kábelt
- Ezzel egy időben hallgatózni kezd, hogy a két alternatív útvonal közül melyiket válassza
- A sebességek, prioritások egyformák, így a MAC dönt: SW3 felé lesz az új RP
- SW4 felengedi a korábban SW3 felé blokkolt portot, további kb. 10-15 mp után a hálózat megjavította magát
- Szemmel láthatóan SW3 a jó döntés, SW5 csak egy felesleges extra ugrás a root switch irányába

Etherchannel

Az STP, a különböző verziók, timerek, implementációk miatt egy bonyolult protokoll, néhány hátránnyal

Egy hátrányára fókuszálunk most, ez egy koncepcionális gyengeség

Az STP de facto azt jelenti, hogy olyan kábelek vannak a rendszerben, amik (ha minden jólmegy) sosem kerülnek használatba (blokkoltak)

Ezek a kábelek pénzbe kerültek (megvenni), időbe kerültek (kiépíteni)

A switch portok amelyekbe bele vannak dugva, szintén pénzbe kerültek

És ott ül kihasználatlanul, miközben a működő kábel esetleg vörösen izzik a nagy forgalomtól

EZ PAZARLÁS

Milyen szuper lenne, ha a ki nem használt kábeleket is valahogy használni lehetne

- Az etherchannel (esetleg port channel) lényege pontosan az, hogy több fizikai kábelt fogjunk egy marokba és
- kezeljük logikailag egy nagy kábelnek (nagy = kétszer, n-szer olyan gyors)
- Amennyiben két, három, akár nyolc fizikai kábelt egy logikai vastag, nagy kábelnek tekintünk, kétszer,
- háromszor, nyolcszor akkora forgalmat vihetünk át rajta
- ÉS! Ha egy kábel hibás lesz, megszakad – akkor a vastag kábel kevésbé lesz vastag, de még mindig
- üzemelni fog és nem vesztünk forgalmat
- Értelemszerűen a kábel mindkét oldalán lévő eszköznek támogatnia kell
- De támogatja is szinte mindenki: minden hálózati eszköz (switch, tűzfal), linux, windows egyaránt
- Szerver oldalról ismerős lehet: bonding, grouping, trunking (ez utóbbi félrevezető, mivel a több VLAN-t
- hordozó portot nevezik a hálózati világban trunk-nek)
- Természetesen itt is több protokollról beszélünk
  - LACP – Link Aggregation Control Protocol (IEEE 802.3ad)
  - PAgP – Port Aggregation Protocol (Cisco)
- Mindkettő ugyanarra való, PAgP csak Cisco eszközökben, LACP viszont bármiben van ahova
- implementálták
- Nyilván a kettő egymással nem kompatibilis (PAgP az egyik oldalon, LACP a másikon nem fog menni)

Gyakorlat - Etherchannel

- Az eddigi hálózat kiegészül egy újabb kábellel:

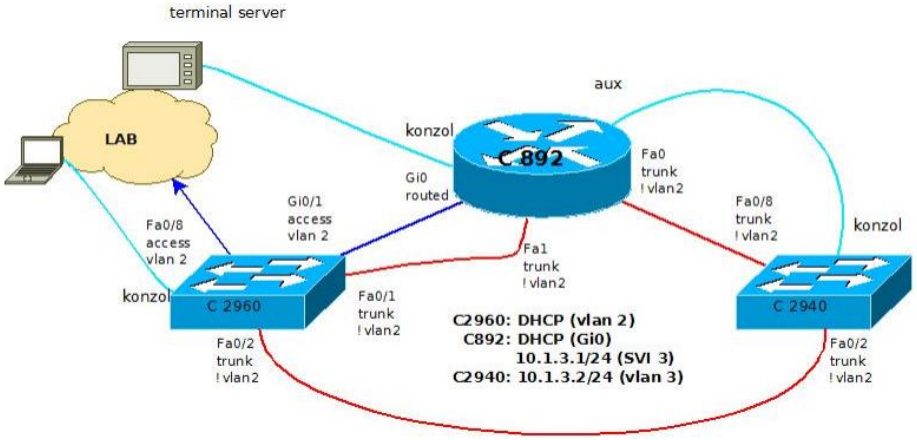
  c2940 és c2960 közé, Fa 0/3 kerül összekötésre

- Ez Fa0/2 -vel fog párban etherchannel-t képezni

Will an EtherChannel Form?

LACP	Active	Passive
Active	Yes	Yes
Passive	Yes	No

PAgP	Desirable	Auto
Desirable	Yes	Yes
Auto	Yes	No

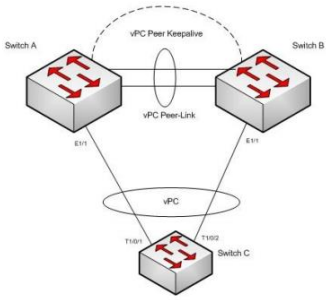


Etherchannel - gondolatok

- A táblázatban láthatóan az alapértelmezett STP cost értékek az egyes interfész sebességekhez
- Érdemes emlékezni rá, hogy egy szint felett már nincs eltérés, a 20Gbit és 40 Gbit linkek között nincs költség különbség
- Érdemes emlékezni arra is, hogy a sebesség nem minden – az 54 mbites wifinek a költsége jobb, mint az 50 mbites drótnak, de valóban jobb?
- Az etherchannel egyik koncepciója a rendelkezésreállás (redundancia) növelése (pl. szerverek, NASok esetén)
- Mégis hogyan növeli a redundanciát, ha az etherchannel mindkét portja ugyanabban a switchben van?
- A switch maga is SPoF
- Kézenfekvő igény lenne az etherchanneleket különböző switchekben végződtetni

- Erre két lehetőség van: stack vagy VPC

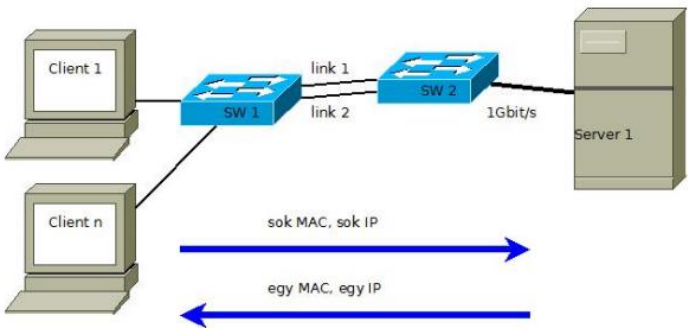
Data rate	STP Cost
4 Mbps	250
10 Mbps	100
16 Mbps	62
100 Mbps	19
1 Gbps	4
2 Gbps	3
10 Gbps	2

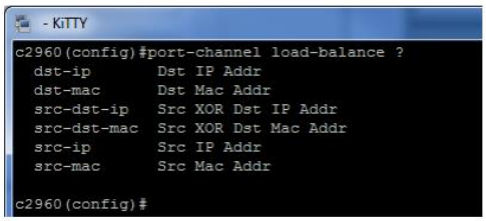


Etherchannel alkalmazása

A terhelés eloszlása:

- A lehetőségek platformtól és operációs rendszertől függnek
- Terhelés elosztás alatt azt értjük itt, hogy milyen módon osztja el a switch, melyik keret melyik porton mehet
- Még a legkisebb switch is támogat több fajta hashing algoritmust, amely eredménye egy port szám amin a keret kimehet
- A döntés alapozható: MAC címre, IP címre, esetleg TCP/UDP portszámra, valamint ezek keverékére
- Figyelembe lehet venni forrás oldalon vagy cél oldalon
- Át kell gondolni, melyik a valóban hasznos, ehhez ismerni és érteni kell a tipikus forgalmi modelleket
- Ha sok kliens forgalmaz ugyanahhoz a szerverhez (sok különböző forrás MAC, ugyanaz a cél MAC)
- A válasz pedig mindig ugyanarról a MAC címről és IP címről jön
- Akkor SW1 -en nincs értelme cél MAC alapú hash-t, SW2 -n pedig nincs értelme forrás MAC alapút választani
- Hacsak nem az aszimmetrikus terhelés a cél (de miért lenne az)





RAID

Bevezető

A soros ATA- (SATA) vezérlők megjelenésének köszönhetően a számítógépbe építhető merevlemezek száma lényegesen megnőtt, és ezzel egy időben a RAID név is beivódott az otthoni felhasználók tudatába. A RAID-kötetek a merevlemezek számától és a beállításoktól függően biztonságosabbá és/vagy gyorsabbá tehetik az adattárolást.

Bár az ASUS és az Abit már a Pentium III processzorok korában gyártott PATA RAID-vezérlős alaplapokat otthoni felhasználóknak, ezek kevésbé terjedtek el. Ennek több oka is volt: az akkori RAID-vezérlők megbízhatatlanok voltak, a merevlemezek ára igen borsos volt, továbbá az operációs rendszerek is kiforratlannak számítottak -- kivéve a Windows 2000 és Windows XP ősatyját, az akkoriban csak szerver- és munkaállomás-feladatokra képes Windows NT-t.

A vezérlő megbízhatósága egy RAID-konstrukcióban létszükséglet, ugyanis bizonyos esetekben a RAID-kötet szétesése vagy összeomlása teljes adatvesztést eredményezhet. Első körben megvizsgáljuk e betűszó jelentését, a kivitelezéshez szükséges eszközöket, majd szemléltetjük az egyes RAID-kötetek tulajdonságait.

A RAID értelmezése

Az IBM már 1978-ban olyan rendszerekkel kísérletezett, amelyek segítségével visszaállítható a tárolóban megsérült adat. Bár a RAID-technológia csak 1988-ban vált valósággá, a korai próbálkozások szolgáltak több RAID-üzemmód alapjául.

A RAID (Redundant Array of Independent – vagy Inexpensive – Disks) betűszó magyar jelentése független – vagy olcsó – lemezek redundáns tömbje. Ha értelmezni szeretnénk a jelentését, több részre kell bontanunk.
Elsőként a redundáns tömb szorul magyarázatra: ez a RAID esetében olyan megtöbbszörözött tárolóegységet jelent, amelyet a rendszer és a felhasználó egyetlen tárnak lát. A független lemez jelentése szinte egyértelmű: a RAID-kötet létrehozásához több – minimum két darab – merevlemez szükséges. A RAID segítségével tehát létrehozhatunk gyors vagy hibatűrő kötetet, de egyes módozatai mindkét tulajdonságot képesek ötvözni magukban.

A RAID-technológiát kezdetben csak szerverekben, illetve nagyobb munkaállomásokban használták, ahol a gyors és megbízható, SCSI-csatolófelületű merevlemezek széles körben meghonosodtak. Az otthoni felhasználók körében részint a már említett merevlemezek magas ára, részint pedig az otthoni felhasználáshoz szükséges RAID-vezérlők kései megjelenése miatt terjedt el lassabban.

Természetesen ők is vásárolhattak volna SCSI-kártyát, de annak ára merevlemezekkel együtt olyan magas volt, hogy egyszerűen nem érte meg beruházni ebbe, így a RAID-technológia előnyei akkor realizálódtak igazán, amikor a merevlemezek már megfizethető közelségbe kerültek, illetve az alaplapok is integrált RAID-vezérlővel felszerelve érkeztek. Bár tény, hogy

ezen időszak előtt már bőven lehetett PCI-csatolófelületű, RAID-képességekkel ellátott IDE-vezérlőkártyákat vásárolni, mégis kevesen éltek a lehetőséggel. A technológia térhódítása valójában a SATA-vezérlők megjelenésével vette kezdetét.

RAID változatok

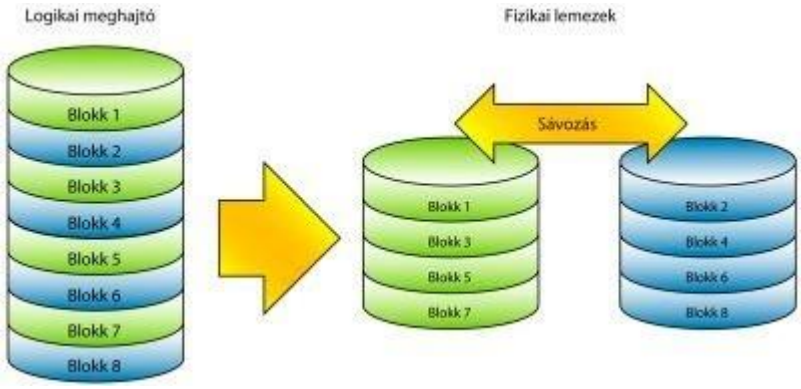
A RAID nem egy konkrét módszer – több alfaja is ismert, illetve vannak olyan, egyes vezérlőkártyák által támogatott változatok is, amelyek nem is tartoznak bele az eredeti specifikációba. Lássuk tehát először, hányfajta RAID-tömböt építhetünk!

JBOD

A JBOD tulajdonképpen nem valódi RAID-kötet, de a RAID-vezérlők többsége felkínálja ezt a lehetőséget. A JBOD (Just a Bunch Of Disks – csak egy köteg lemez) megoldás lényege, hogy több merevlemezt egynek lásson az operációs rendszer, de ehhez nem használ redundanciát; szimplán összefűzi a merevlemezeket, tehát a lemezek kapacitása összeadódik, ettől fogva egyetlen nagy lemeznek látszanak. Komoly hátránya, hogy ha az egyik meghajtó meghibásodik, minden adatunk elvész. Használata éppen ezért csak korlátozottan ajánlott, fontos állományok tárolására inkább ne vegyük igénybe.

RAID 0 -- Striping (Csíkozás, sávozás)

Az első valódi RAID-mód kivitelezéséhez legalább két merevlemez szükségeltetik. E konstrukció előnye a nagy sebesség, amelyet úgy ér el, hogy a fájlok tartalmát egyenletesen elosztja a merevlemezek közt. Ennek hála a két lemez kapacitása összeadódik, és csak egy egységnek látszik a felhasználó felé.



RAID 0

Amennyiben egy állomány mérete meghaladja egy RAID blokk méretét, szétdarabolódik, és az így felszeletelt állomány már eloszlik a merevlemezek közt. Tegyük fel, hogy a RAID blokk mérete 32kbyte. Egy nagyobb fájl 0. sorszámú (tehát első) 32k-s darabja az első merevlemezen, az 1. sorszámú 32k-s darabja a második merevelmezen, a 2. sorszámú 32k-s darabja ismét az első merevlemezen tárolódik, és így tovább. Az ily módon készített RAID-tömb tehát olyan merevlemeznek látszik, amelynek sávonkénti szektorszáma megtöbbszöröződik a tömbben részt vevő egységek számának függvényében, így a rendszer hosszabb adatblokkokat tud írni és olvasni fejmozgatás nélkül. Természetesen a

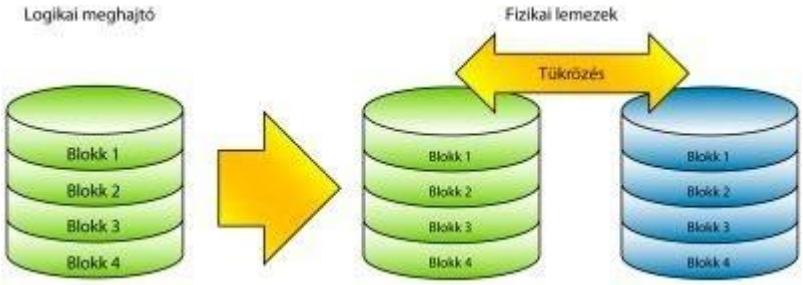
merevlemezek gyorsítótárai is összeadódnak, ennek eredményeképp tovább nőhet a sebesség. Ez az oka annak, hogy ideális körülmények között az írás és olvasás sebessége megtöbbszöröződhet. Azért említünk ideális körülményeket, mert a csíkozás mérete erősen befolyásolja a lemezek teljesítményét.

Ha túl kicsi méretet választunk a RAID 0 tömbben, akkor egy nagy állományt sok apró részre kell bontani, ennek kiírása pedig – mivel a vezérlő miatt ez pluszidőbe telik – nem növeli számottevően a teljesítményt. Túl nagy blokkot választani szintén rossz döntés, mert sok apró állomány esetében a blokk mérete meghaladhatja az állományét, amely így nem darabolódik, hanem elfér egy tömbben is, tehát ez esetben egyszerre csak egy lemez dolgozik. Az ideális RAID blokkméret 32 vagy 64 kilobájt, manapság talán az utóbbi bizonyul jobb választásnak.

A RAID 0 legnagyobb hátránya a biztonság hiánya. Mivel az adatok két vagy több merevlemezen szétdarabolva helyezkednek el, már egy lemez kiesése is teljes adatvesztéssel jár, és ezek az adatok már semmilyen módszerrel nem állíthatók vissza.

RAID 1 – Mirroring (tükrözés)

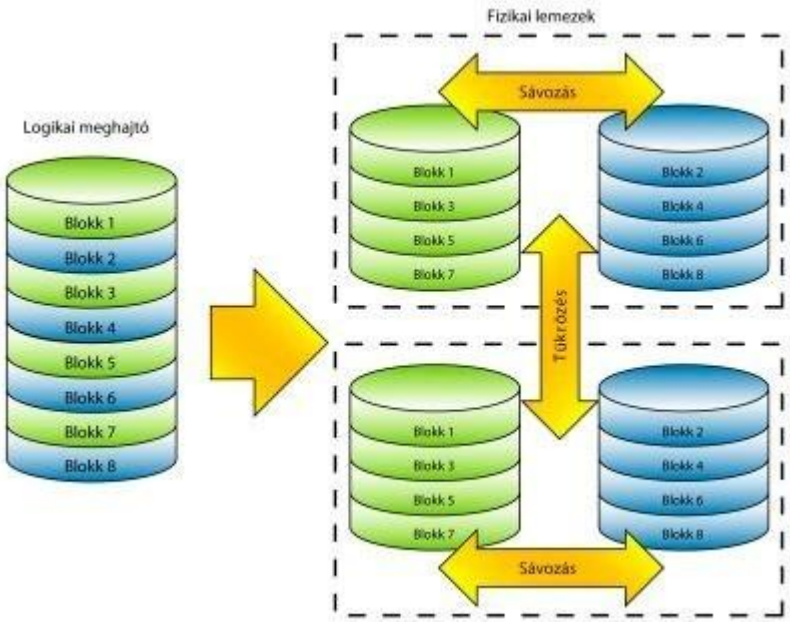
A legbiztonságosabb, de majdnem a legnagyobb fajlagos költségű megoldás. A RAID 0-hoz hasonlóan ez esetben is minimum két merevlemezre van szükség. A RAID 1 konstrukcióban a lemezek egymás tükörképei, tehát mindegyik lemez tartalma bitről bitre megegyezik. Ez az oka a magas fajlagos költségnek, hiszen csak egylemeznyi tárkapacitást tudunk megtölteni adatokkal, a többi csupán tükörképet tárol, vagyis nem jelent a felhasználó számára szabadon hasznosítható tárterületet. Sebességét tekintve a RAID 1 semmivel sem gyorsabb, mint ha csak egy merevlemezt alkalmaznánk. Igaz, a professzionális vezérlők képesek több lemezről egyszerre olvasni, otthoni környezetben azonban ezt nélkülöznünk kell. Amennyiben a tömbben található egyik merevlemezünk „kipukkadna”, adataink a többin még megmaradnak.



RAID 1

RAID 0+1 – Striping + Mirroring (Csíkozás és tükrözés):

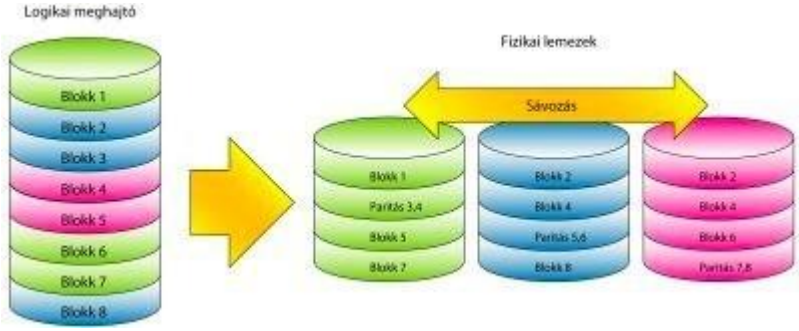
Ez a megoldás ötvözi a RAID 0 gyorsaságát a RAID 1 megbízhatóságával, ám ez sem költségkímélő, ráadásul helykihasználása sem nevezhető optimálisnak. Képességeinek kiaknázásához legalább négy merevlemez szükségeltetik, ebből azonban csak kettőnek használható fel a tárterülete.



RAID 0+1

RAID 5

A RAID következő lépcsőfoka, amely széles körben terjedni kezdett. Valójában a RAID 3 és RAID 4 továbbfejlesztett változata, amelyek egyébként nem terjedtek el, de az alapkoncepció a RAID 0 sávozásából ered. A RAID 5 kiaknázásához legalább három merevlemez szükséges, ebből kettő tárterületét tudjuk felhasználni.



RAID 5

A RAID 5 különlegessége, hogy gyors és biztonságos. A gyorsaság -- mint említettük -- a sávozásból adódik, míg a biztonság alapja a paritás, mely a lemezeken szétszórva helyezkedik el. A paritás olyan redundáns adatokat tartalmaz, amelynek segítségével az eredeti állomány bármely lemez hibájának esetén visszaállítható a másik kettőn tárolt másolat alapján. Ha háromból egyszerre két egység is meghibásodik, az adatok elvesznek, de ennek valószínűsége külső behatás -- például beázás, villámcsapás -- nélkül elég alacsony. A RAID 5 gyors és biztonságos, ezt javasoljuk leginkább, ha RAID-kötetet szeretnének otthonra.

Egyéb RAID megoldások

Három fontos RAID-típust kell megemlíteni ahhoz, hogy megértsük a RAID 5 előmenetelét: RAID
2, RAID 3 és RAID 4. Mint említettük, ezek „reinkarnációja”, illetve tökéletesített változata a RAID 5. Közös tulajdonságuk, hogy a paritásadatokat nem szétszórva, hanem egy külön paritáslemezen tárolják. A RAID 2 nem honosodott meg, mert sok paritáslemez kellett hozzá. A továbbfejlesztett RAID 3 már csak egy pluszlemezt igényelt, valamint az állományok tárolása bájtonként történt, szemben a RAID 2 bitenkénti tárolásával. A RAID 4-től már csak egy lépés volt a RAID 5. Bár a fix paritáslemezre itt is szükség van, az adatok már sávozással kerülnek a lemezekre.

Létezik még néhány RAID-típus, azonban ezek tárgyalásába nem mennénk bele, mert nem javasoltak otthonra: egyrészt drágák, másrészt megfizethető vezérlő hiányában úgysem tudjuk kihasználni őket. Ilyen például a RAID 6 és 7, amelyek a RAID 5 továbbfejlesztett változatai, illetve néhány cég egyéni próbálkozása, mint például a RAID 1.5.

RAID 10 és a Matrix RAID

Az Intel ICHXR (az „X” jelöli a lapkakészlet számát, az „R” pedig a RAID-képességet) déli hidak általában lehetőséget kínálnak a RAID 10 és Matrix RAID technológia kihasználására is, bár ez utóbbi nem hivatalos RAID-eljárás.

A RAID 10 a RAID 0+1 fordítottja. Bár kivitelezése kissé eltérő, a gyakorlatban ugyanazok a képességei, mint a RAID 0+1-nek, tehát hasonlóan pazarló módon bánik a felhasználható tárterülettel. Nem érdemes használni.
A Matrix RAID az ICH-vezérlőktől függően különböző fokozatokat kínál. A régebbi ICH6R sorozat a Matrix technológiával karöltve lehetővé teszi, hogy négy lemez helyett kettőn is létrehozhassunk RAID 0+1 kötetet. A Matrix eljárással a két merevlemez két külön részre osztható -- az ICH6R esetében egy sávozott és egy tükrözött részre --, és az adatok keresztbe tárolhatók. Az egyik lemez sávozott részének tükörképét a másik lemez tükrözött részén -- tehát keresztbe -- tárolja, így ha az egyik lemez tönkremegy, az adatok nem vesznek el, hiszen a másikon ott a tükörképe.

Az újabb, ámde nem a legújabb ICH7R a RAID 5 képességeit bővíti ki, és az eddig még (magas ára miatt) nem említett RAID 50 módot alkalmazza négy merevlemezen. A RAID 50 a RAID 0 és RAID 5 egyesítése, de az imént bemutatott Matrix technológiát alkalmazva sokkal kevesebb lemez szükséges hozzá.

Hozzávalók, hardveres, szoftveres és hibrid megoldások

Eddig kizárólag a vezérlőkártyás kiépítésekről esett szó, ezek azonban nem mindig teljes értékű hardveres megoldások, a hibrid kártyák vezérlését a rajtuk tárolt szoftverek biztosítják. A valódi hardveres kártyák ára nagyon magas, otthoni használatra nem érdemes

ilyet vásárolni. A lényegi különbség a hardveres és a hibrid megoldások között, hogy míg utóbbiak a számítógép központi processzorát terhelik, előbbiek teljes mértékben tehermentesítik azt. Nagy teljesítményt igénylő szerverkörnyezetben -- ahol sok merevlemez van -- a hardveres vezérlő jelenti az egyetlen megoldást, ráadásul a professzionális kártyák nemcsak 2–4–6 merevlemezt képesek kiszolgálni, hanem annál sokkal többet is.

A hibrid megoldások között találunk olyan vezérlőket is (főképp Intel lapkakészletű alaplapokban), amelyek már a RAID 10-et is tudják, azonban léteznek olyanok is, amelyeknél a RAID 0+1 jelenti a csúcsot. Ez esetben mindenképpen érdemes a vezérlőt gyártó cég honlapját felkeresni, mert a RAID-vezérlő BIOS-szoftvere az alaplap BIOS-szoftveréhez hasonlóan frissíthető, így előfordulhat az is, hogy egy újabb BIOS-szal bővül a választék.

A RAID-kötethez több eszközre van szükség -- elsősorban egy RAID-képességekkel felvértezett SATA vagy PATA IDE-merevlemez-vezérlőre (a SCSI-t is ide sorolhatnánk, de mi most kizárólag az otthoni felhasználók számára szükséges legegyszerűbb változatokat tekintjük át), amely lehet az alaplapra integrálva vagy külön kártya formáját is öltheti. Attól függően, hogy milyen típusú tömbö(ke)t szeretnénk létrehozni, annyi csatolóval ellátott vezérlőre lesz szükségünk. A vezérlőn lévő csatolók száma határozza meg a hozzá csatlakoztatható merevlemezek maximális számát is. Mint azt már bemutattuk, legalább két, három vagy négy merevlemez kell hozzá annak függvényében, hogy melyik változatot szeretnénk használni. Fontos hangsúlyozni, hogy az optimális teljesítmény érdekében ajánlott négy teljesen egyforma merevlemezből kiépíteni a RAIDkötetet. Amennyiben eltérő méretű lemezeket fűzünk össze, a RAID-kötet méretét a legkisebb tárterületű lemez határozza meg, a többi hely parlagon marad.

További technikák

Hot Swap

Mivel a RAID lényege a hibatűrés – igaz, ez alól kivétel a RAID 0 – a Hot Swap szolgáltatás a RAID egyik legfontosabb eleme. Ez lehetővé teszi a meghibásodott merevlemezek cseréjét a számítógép kikapcsolása nélkül, akár írás vagy olvasás művelet közben is. Ehhez természetesen az operációs rendszer és a RAID-vezérlő támogatása is szükséges.

A Hot Swap szolgáltatást már a Windows 2000-ben is előkészítették, ám az XP sokkal jobban kezeli. A vezérlők terén már nem ilyen egyértelmű a helyzet. A régi, főleg első generációs SATAvezérlők nagy része még nem tűri a menet közbeni meghajtócserét – általában nem ismerik fel az újonnan behelyezett merevlemezt –, és lefagyással reagálnak. PATA-vezérlőnél sem érdemes a menet közbeni csereberéléssel kísérletezni, mert a hagyományos, párhuzamos ATA-merevlemezek nem viselik jól az effajta kezelést. Kísérletezni persze lehet, de ezt még akkor próbáljuk meg, amikor semmilyen fontos adat sincs RAID-tömbünkben.

Ha menet közbeni cserére számítunk, lemezeinket érdemes merevlemezfiókba telepíteni, a profi rendszerek is mind ilyenek. Ha nem fiókokban tároljuk lemezeinket, akkor se

csüggedjünk – viszont biztos kézre és higgadt mozdulatokra lesz szükségünk. Ha a SATA-merevlemezt fiók nélkül helyeztük a gépünkbe, menet közbeni cseréjekor feltétlen tartsuk be az itt meghatározott sorrendet: elsőként a merevlemez tápcsatlakozóját csatlakoztassuk, majd a tányérok felpörgése után az adatkábelt is. Abban az esetben, ha a vezérlő nem ismerné fel azonnal az egységet, indítsuk el az Új hardver felismerése funkciót a Windows Eszközkezelőjében.

A Hot Swap csere természetesen RAID 1 vagy RAID 5 kötetek esetében lehetséges, ugyanis a hibás merevlemezt csak e tömbökben lehet működés közben büntetlenül kicserélni.

Hot Spare

A Hot Spare módszer olyan megoldás, amely a RAID-konstrukcióban felhasználói beavatkozás nélkül azonnal helyettesíti a hibás merevlemezt, tehát automatikusan újraépíti és kijavítja a RAIDkötetet. Ennek feltétele egy kihasználatlan, kifejezetten erre a célra fenntartott tartalék merevlemez megléte. A hiba felléptekor a rendszer használatba veszi e tartalék lemezt, konfigurálja és átmozgatja rá a szükséges adatokat. Ha ez megvan, a meghibásodott egységet leválasztja a rendszerről, majd valamilyen módon – a Windows driveren keresztül vagy a következő BIOSbejelentkezéskor – értesíti a felhasználót. A meghibásodott egységet jóra cserélve az lesz a következő tartalék lemez.

A PPP müködése

Soros kommunikáció

Keretek bitjei egymást követik a fizikai közegen –soros átvitelt alkalmaznak

Fizikai közegen alkalmazható jelölések

- Non-return to zero level (NRZ-L)
- High Density Binary (HDB3)
- Alternate Mark Inversion (AMI)
- A jelölések szerepe ugyanaz, mint Ethernet esetén a Manchester kódolásé

Soros kommunikációs szabványok

- RS-232-E
- V.35
- HSSI

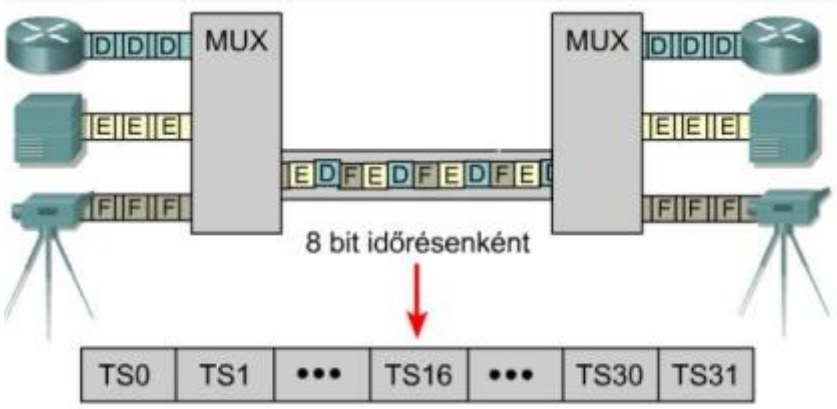
Időosztásos multiplexelés (TDM –Time Division Multiplexing)

Több információforrás adatait is egyetlen közös csatornán, vagyis jelzéssel lehet továbbítani majd visszaállítani a az eredeti adatfolyamot

Jellemzők

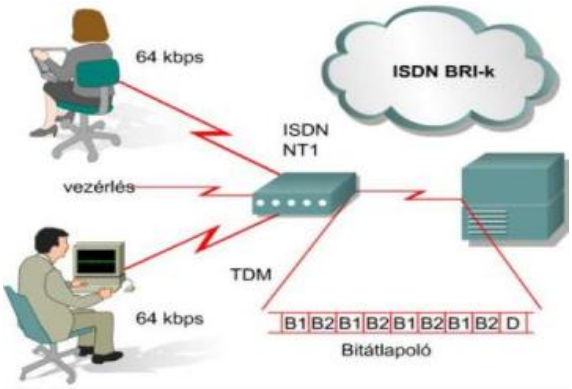
- Több forrás adatát is ugyanaz a csatorna továbbítja
- A kimeneti csatorna időszeletenként vehető igénybe
- Az időrések mindig rendelkezésre állnak (akkor is, ha nincs adat)
- Egyszerre egy bit vagy egy bájt átvitele történik meg
- A bemeneti csatornák eltérő kapacitásúak is lehetnek–Statikus sávszélesség

Fizikai rétegbeli eljárás, független a bemeneti csatorna második rétegbeli protokolltól

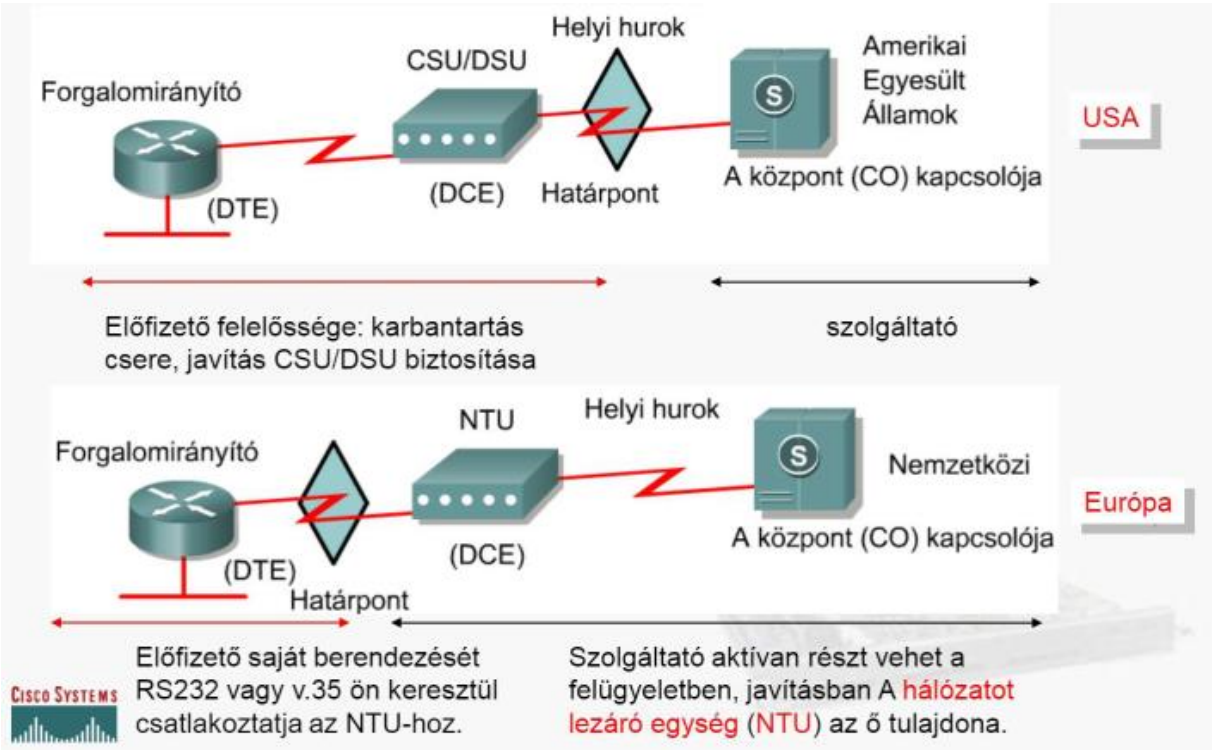


TDM példa

- ISDN BRI (két 64kbps B csatorna, egy 16kbps D csatorna)
- Kilenc időrés használata(B1-B2-B1-B2-B1-B2-B1-B2-D)
- Bitátlapoló TDM



Határpont



DCE/DTE

Soros összeköttetések végénél található eszközök

- Adatkommunikációs berendezés (DCE –Data Communications Equipment)
- Adat végberendezés (DTE –Data Terminal Equipment)

DCE eszközök közötti kapcsolatot a WAN szolgáltató biztosítja

DCE/DTE szabványok által meghatározott paraméterek

- Mechanikai és fizikai specifikáció (érintkezők, csatlakozók, stb.)
- Elektromos jellemzők (feszültségszintek)
- Funkcionalitás (interfészek jelentései, funkciói)
- Procedurális leírások (adattovábbítás eseményeinek meghatározása)

Cisco által bevezetett DCE/DTE csatlakozás: Smart Serial

HDLC

1979-ben az ISO elfogadta a HDLC-t. Szabványos,bit alapú, szinkronsoros összeköttetéseken beágyazást végző adatkapcsolatirétegbeli protokoll. Cisco alapértelmezett protokoll

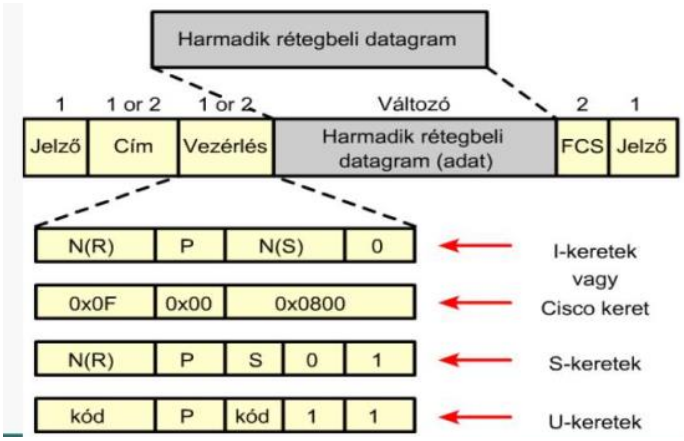
Származtatott protokollokatkapcsolatelérési protokolloknak hívjuk:

- Link Access Procedure, Balanced (szimm. kapcsolatelérési eljárás, LAPB) az X.25-höz
- Link Access Procedure az ISDN D csatornák számára (LAPD)
- Link Access Procedure for Modems (LAPM) és PPP a modemekhez
- Link Access Procedure for Frame Relay (LAPF) a Frame Relay összeköttetésekhez

A HDLC szinkron soros átvitelt alkalmaz, két pont között hibamentes átvitelt biztosít.Nyugtákkal, ablakozással biztosítja az adatfolyam-vezérlést, hibajavítást.

HDLC kerettípusok

- Információs keret (I-Frame)
  Átviendő adatokat tartalmazza
- Felügyeleti keret (S-Frame)
  Kérés/válasz eljárások amikor adat-ráültetés nem történt
- Számozatlan (U-Frame)
  További összeköttetés-vezérlő funkciókat biztosít



Soros interfész hibaelhárítása

Serial 0/0	Line protocol	Leírás
up	up	Normál működés
down	down	Nincs vivőjel, helytelen kábelezés, nincs órajel DCE oldalon, hardverhiba
up	down	Hibás konfiguráció, nincs keepalive, zajos vonal, időzítési probléma
up	up (looped)	Az áramkörben hurok (loopback) keletkezett
up	down (disabled)	Túl magas hibaarány, hiba a CSU/DSU egységben, interfészhiba
administratively down	down	Konfiguráció alapján az interfész lekapcsolásra került

```
Router#show interfaces Serial
Router# show controllers Serial
Router#show controllers cbus
Router#debug serial interface
Router#debug arp
```

- **debug serial interface**–Ellenőrzi, hogy a HDLC ébrenléti üzenetek sorszáma nő-e. Ha nem, akkor valószínűleg időzítési problémák léptek fel az interfészkártyánál vagy a hálózaton.
- **debug arp**–Jelzi, hogy a forgalomirányító küld-e adatokat más forgalomirányítókról, illetve ismer-e meg ilyeneket (ARP-csomagok segítségével) a WAN-felhő másik oldaláról. A parancsot akkor kell használni, ha a TCP/IP hálózat bizonyos csomópontjai válaszolnak, mások viszont nem.
- **debug frame-relay lmi**–LMI (Local Management Interface, helyi kezelőfelület) információkat szerez be, amelyek alapján megállapítható, vajon egy adott Frame Relay kapcsoló és egy forgalomirányító küld és fogad-e LMI-csomagokat.
- **debug frame-relay events**–Megállapítja, hogy van-e adatcsere egy adott forgalomirányító és Frame Relay kapcsoló között.
- **debug ppp negotiation**–Megjeleníti a PPP-beállítások egyeztetésére szolgáló, a PPP indítási folyamat során továbbított PPP-kereteket.
- **debug ppp packet**–Megjeleníti az elküldött és a fogadott PPP-kereteket.Ez a parancs alacsony szintű kerettartalom-kiíratást végez.
- **debug ppp**–Megjeleníti a PPP-kapcsolat beállításainak egyeztetésével és működésével összefüggő hibákat, mint például ha nem megengedett vagy hibás keretek érkeznek.
- **debug ppp authentication**–Megjeleníti a PPP CHAP (Challenge Handshake Authentication Protocol, kihívásos kézfogás hitelesítési protokoll) és a PAP (Password Authentication Protocol, jelszóhitelesítő protokoll) által elküldött kereteket.

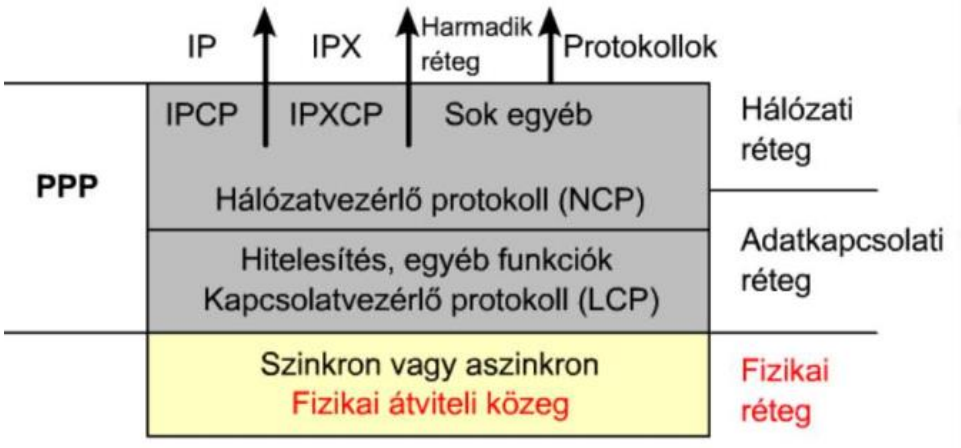
A PPP réteges architektúrája

Kapcsolatvezérlő protokoll (Link Control Protocol, LCP)

- Feladata a pont-pont összeköttetés létrehozása.
- konfigurálása és tesztelése
- a WAN-kapcsolat beállításainak egyeztetése és kézben tartására.

Hálózatvezérlő protokoll(Network Control Protocol, NCP)

- Feladata a különböző hálózati rétegbeli protokollok konfigurálása.
- protokollok beágyazása és a rájuk vonatkozó beállítások egyeztetése.



PPP architektúra

Kapcsolatvezérlő protokoll (LCP)

- wAN kapcsolat beállítása és folyamatos egyeztetése–Hitelesítés, tömörítés, hibafelismerés
- Multilink (többrésztvevős PPP a terheléselosztás biztosítására)
- PPP visszahívás, callback
- Csomagméret korlátozások kezelése, beállítási hibák felfedezése, kapcsolat megszakítása, helyes és hibás működés felismerése

Hálózatvezérlő protokoll (NCP)

- Hálózati rétegbeli protokollok beágyazása, beállításuk egyeztetése
- Több protokoll egyidejű támogatása
- Minden hálózati protokollhoz külön vezérlőprotokoll tartozik (pl. IPCP, IPXCP, stb.)
- Hálózati protokoll azonosítás szabványos kódokkal

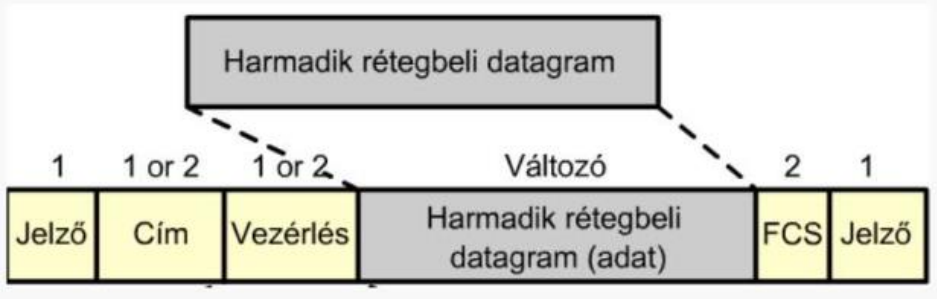
A PPP a következő típusú fizikai interfészeken használható:

- Aszinkron soros Szinkron soros
- Nagysebességű soros interfész (High Speed Serial Interface, HSSI)
- Integrált szolgáltatású digitális hálózat (ISDN)

Hitelesítés–A hívó félnek hitelesítési információt kell megadnia, hogy rendelkezik-e a rendszergazda engedélyével a híváshoz. A forgalomirányítók hitelesítési üzeneteket küldenek egymásnak. Két lehetséges megoldás a hitelesítés elvégzésére a jelszó-hitelesítő protokoll (PAP)és a kihívásos kézfogás hitelesítésiprotokoll (CHAP) Tömörítés–a PPP-kapcsolatok átviteli sebességének növelésére használhatók. A keretek kibontását a túloldalon a protokoll végzi. A Cisco forgalomirányítók két tömörítő protokollt ismernek, ezek a Stacker és aPredictor. Hibafelismerés–A Quality (minőség) és a Magic Number (~bűvös szám) funkciók segítségével megbízható, hurokmentes adatkapcsolat hozható létre. Multilink–A Cisco IOS 11.1-es és újabb kiadásai a multilink (többrésztvevős) PPP-t is támogatják. Segítségével terheléselosztás valósíthatómeg a PPP által használt

forgalomirányító-interfészek között. PPP visszahívás–A biztonság növelése érdekében a PPP feletti visszahívást is támogatják. Visszahívási ügyfélként és visszahívási kiszolgálóként egyaránt használható. Visszahívásnál az ügyfél egy hívással létrehozza a kezdeti kapcsolatot, kéri a kiszolgálótól a visszahívását, majd megszakítja a kapcsolatot. A visszahívó forgalomirányító válaszol az első hívásra, és a konfigurációjában megadottak alapján visszahívja az ügyfelet.

- Jelző–A keret elejét és végét jelzi, a bináris 01111110 sorozatból áll.
- Cím–A szokásos szórási cím, mely a bináris 11111111 sorozat. A PPP nem használ egyedi állomásazonosítókat.
- Vezérlés–1 bájt, amely a bináris 00000011 sorozatot tartalmazza. Azt jelenti, hogy a felhasználói adatokat számozatlan keretekben kell továbbítani. Mindez az 1-es típusú logikai kapcsolatvezérléshez (LLC) hasonlóan nyújtósszeköttetés-mentes szolgáltatást.
- Protokoll–2 bájt, a keret adat mezőjébe beágyazott protokollt azonosítja.
- Adat–0 vagy több bájt, amely a protokoll mezőben megadott protokoll datagramját tartalmazza. Az adat mező végét úgy lehet megtalálni, hogy akeret végéről leválasztjuk a záró jelzőbiteket és a 2 bájtos keretellenőrző(FCS) mezőt. Az adat mező maximális hossza alapértelmezésben 1500 bájt.
- FCS–Általában 16 bit, vagyis 2 bájt, hibaellenőrzési célból kerül csatolásra a kerethez.



PPP kapcsolat felépítése

Összeköttetés felépítése

- Összeköttetés létrehozása
- Hitelesítés (elhagyható)
- Hálózati rétegbeli protokoll használat

PPP kapcsolatok kereteinek osztályozása

- Kapcsolat-felépítő keret-az összeköttetés felépítésére és konfigurálására használjuk.
- Kapcsolat-bontó keret-feladata az összeköttetés lebontása
- Kapcsolat-fenntartó keret-az összeköttetés kezelésére és hibaelhárításra alkalmasak.

PPP összeköttetés lezárása

- addig marad adatcserére konfigurálva, míg valamelyik esemény be nem következik

- LCP vagy NCP keretek lezárják az összeköttetést
- Inaktivitási időmérők egyike lejár–Felhasználó beavatkozik

Összeköttetés felépítése

Összeköttetés létrehozása

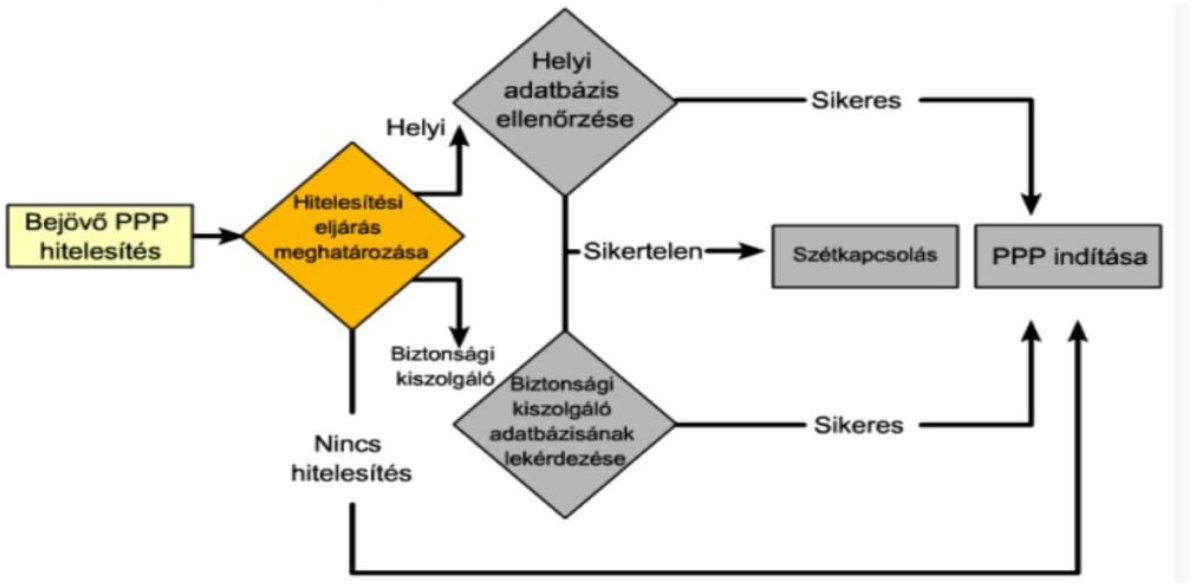
- LCP figyelés, adatkapcsolat konfigurálása
- LCP nyitás, kapcsolat létrehozása, beállítások egyeztetése
- Konfiguráció nyugtázó keret küldése, fogadása

Hitelesítés (elhagyható)

- Résztvevő felek hitelesítése
- Összeköttetés minőségének ellenőrzése (LCP)

Hálózati protokoll használata

- NCP csomagok küldése
- Hálózati rétegbeli protokoll(ok) kiválasztása (például IP), konfigurálása
- Hálózati rétegbeli protokoll(ok) csomagjainak továbbítása



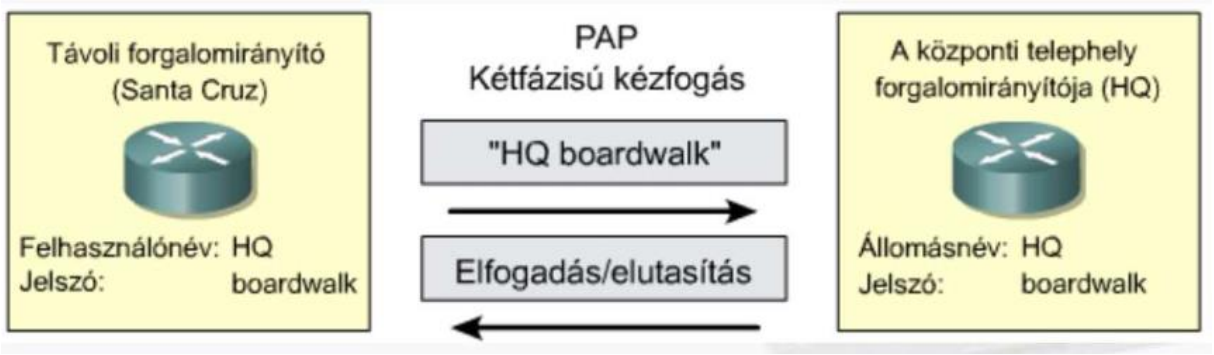
Az encapsulation pppparancs kiadásakor dönthetünk PAP vagy CHAP hitelesítés mellett. Ha nincs semilyen, akkor a PPP-kapcsolat azonnal létrejön. Ha előírtunk valamit, akkor a következő lépésekre kerül sor. A felek egyeztetik a hitelesítési eljárást.

PAP hitelesítés

Hálózat biztonság –jelszóhitelesítő protokoll alkalmazásaPassword Authentication Protocol, PAP

Jellemzők

- A jelszó egyszerű szövegként kerül továbbításra
- A behívó állomás vezérli a próbálkozásokat
- Nem erős hitelesítési protokoll



CHAP hitelesítés

Challange Handshake Authentication Protocol, CHAP
kihívásos kézfogás hitelesítési protokoll

A háromfázisú kézfogás

- az összeköttetés létrehozásakor,
- másrészt a kapcsolat fennállása alatt folyamatosan ellenőrzi a távoli állomás személyazonosságát.
- A helyi forgalomirányító kihívó üzenetet továbbít a távoli állomásnak.
- Válaszként egy egyirányú hasító függvény –általában Message Digest 5, röviden MD5 –segítségével számított értéket küld vissza. (a jelszó és a kihívóüzenet alapján)
- Helyi forgalomirányító összeveti a választ az általa számított kivonattal. Ha egyezik, akkor sikeres az ellenőrzés, egyébként azonnal bontja a kapcsolatot.

Frame Relay – Virtulái körök

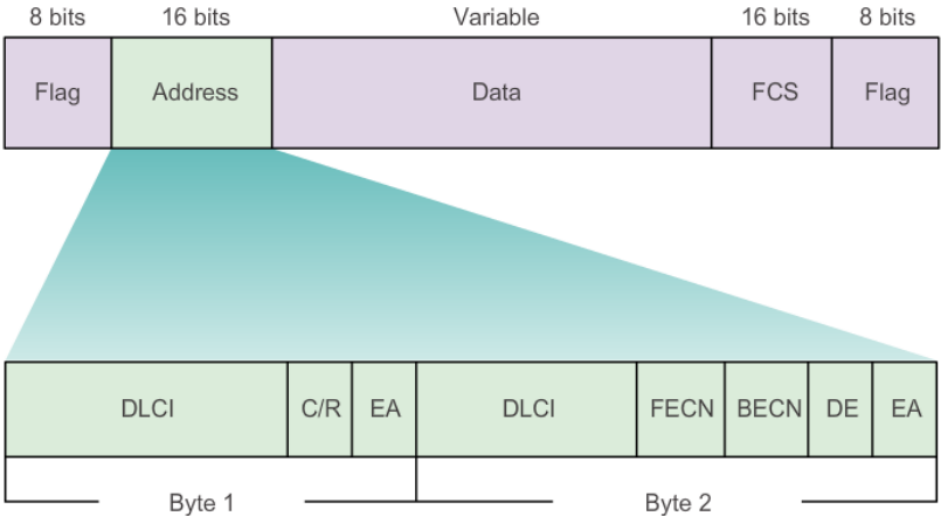
Az X.25 tervezésekor az átviteli közegek rossz minôségû analóg vonalak voltak, ez indokolja a többszörös hibajavítást és forgalomszabályzást. A Frame Relay-t azonban jó minôségû ISDN vonalak feletti mûködésre találták ki, elsôsorban az X.25-nél nagyobb sebességigények kielégítésére. Különbözô javaslatok kerültek benyújtásra az ITU-T-hez 1984-ben, az ANSI is foglalkozott a témával a T1S1 bizottságban. A munka eredményeképp megszületett a szolgáltatás specifikációja (I.233) és a LAPD adatkapcsolati réteg egy módosított verziója (Q.922). A Frame Relay történetének lényeges eseménye volt, mikor 1990-ben gyártók megalakították a Frame Relay Fórumot, mely kidolgozott egy a korábbi Frame Relay-el kompatibilis specifikációt, majd azt jelentôsen kibôvítette, hogy megfeleljen a hálózatok összekötésekor felmerülô igényeknek. Ez látszik ugyanis a Frame Relay fô alkalmazási területének.

A Frame Relay csomagkapcsolt adattovábbítási szolgáltatást nyújt egy a hálózat és a felhasználó közötti interface-n keresztül, csakúgy, mint az X.25. A DTE, DCE és VC elnevezések a Frame Relay terminológiában is használatosak. Az X.25-tôl azonban jelentôsen eltér mind formátumában, mind felépítésében. A jelenlegi digitális (gyakran optikai) átviteli vonalak sokkal megbízhatóbbak, mind azok, melyekre az X.25-öt méretezték, így nincsen szükség olyan kifinomult módszerekre, a hibakezelést nyugodtan a felsôbb rétegekre hagyhatjuk, jelentôsen nagyobb teljesítményt és egyszerûbb berendezéseket kapva. A Frame Relay ezen elvek mentén készült és bár tartalmaz CRC mezôt a hiba detektálására, semmilyen a hibás adat korrigálására vonatkozó eljárás (újraküldés) nem része.

Másik fontos különbség a Frame Relay és az X.25 között, hogy a Frame Relay-ben nincsen minden VC-re külön-külön explicit forgalomszabályozás. Most, hogy sok magasabb szintû protokoll tartalmaz ilyen funkciót, erre az adatkapcsolati szinten nincs szükség. A Frame Relay mindössze egy nagyon egyszerû torlódásjelzô megoldást alkalmaz, amivel a hálózat jelzi, hogy közel jár a torlódáshoz. Ezt a jelzést a magasabb szintû protokollok felhasználhatják saját forgalomszabályozásukhoz.

A Frame Relay tulajdonképpen „diétás X.25", úgy is tekinthetjük, mintha az X.25-bôl kiemelték volna a hálózati réteget. A Frame Relay implementációk a dolgozat írásakor még csak a PVC-k használatát támogatták, az SVC kiépítés még vita tárgyát képezte.

A Frame Relay keret roppant egyszerû, az adaton kívül csupán a VC azonosítója található meg benne (változó hosszú lehet, jelenleg 10 bites az elterjedt), valamint a hibadetektáláshoz szükséges CRC és 3 bit, melyeket a forgalomszabályozáshoz használnak.



A VC azonosítót itt DCLI-nek nevezik, (Data Link Connection Identifier), funkciója teljesen megegyezik az X.25 LCI-jével.

A forgalomszabályozás 3 bitje a következô:

FECN (Forward Explicit Congestion Notification): Ennek a bitnek a beállításával jelzi a Frame Relay hálózat, hogy a keret haladási útvonalán valahol torlódás van. Ezt a bitet észlelve a vett keretekben az adott állomás kérheti kommunikációs partnerétôl az adás lassítását. Használata akkor hasznos, ha a magasabb szintû protokollban a vevô végzi a forgalomszabályozást (lassító csomagoknak az adóhoz küldésével), mert ôhozzá futnak be a FECN-nel megjelölt keretek.

BECN (Backward Explicit Congestion Notification): Ezt a bitet torlódás esetén a hálózat a „szembe" jövô keretekben állítja be, melyek a torlódást okozó forrás felé haladnak. Ezt a bitet észlelve a vett keretekben az adott állomás lelassíthatja adását. Használata akkor hasznos, ha a magasabb szintû protokollban az adó végzi a forgalomszabályozást.

DE (Discard Eligibility): Ez a bit jelzi a hálózatnak, hogy az adott keret nem olyan fontos, mint a többi és ha keret eldobására van szükség, akkor az ilyen kereteket dobjuk el elôször. Ez egy nagyon egyszerû prioritásvizsgálatot tesz lehetôvé. Tipikusan akkor állítjuk be ezt a bitet, ha torlódás alakul ki.

A Frame Relay hálózat és az elôfizetô a kapcsolat kiépítésekor megállapodnak, hogy az elôfizetô mennyi adatot kíván a kapcsolaton továbbítani. A hálózat így jobban megtervezheti belsô mûködését, valamint a számlázás is egyszerûbbé válik. A megállapodás 3 értékre terjed ki.

Comitted Information Rate (CIR), ami a hosszútávú átlagos adatsebesség bit/s-ban. Egy megadott idôszakon belül kell tartani ezt az átlagot, az idôszak egyes részeiben ennél magasabb, más részeiben ennél alacsonyabb is lehet az adatsebesség.

Comitted Burst (Bc), ami azt az adatmennyiséget jelenti, amit a hálózat egy adott Tc idô alatt a hálózat átvisz. A CIR ilyen módon kiadódik a Bc és a Tc hányadosaként.

Excess Burst (Be), ami az a maximális adatmennyiség, amit Tc idô alatt elküldve a hálózat még megkísérel továbbítani. Ez egy a Bc-nél nagyobb mennyiség és ha a forgalom a Bc és a Be közé emelkedik, a Bc fölötti forgalmat a hálózat megjelölheti a DE bittel. A Be fölötti forgalom még ennyi garanciát sem kap a továbbításra.

Mindenesetre az megállapítható, hogy a CIR és a DE bit használata szolgáltatóról szolgáltatóra változik, a fenti pontok csak egy lehetséges, bár elterjedt értelmezést adnak.

A Frame Relay Fórum által specifikált kiegészítések neve LMI (Local Management Interface). Egy dedikált VC szolgál az LMI üzenetek adására és vételére, ennek DCLI száma 1023, ezen a PVC-n csak az LMI forgalom bonyolódik. Az LMI keretek segítségével lekérdezhetjük a rendelkezésre álló PVC-k állapotát és a hozzájuk tartozó DCLI-t, multicast funkciókat vehetünk igénybe (1019-1022-ig a DCLI-k multicast csoportokat jelentenek) és forgalomszabályozást is végezhetünk, ha erre a magasabb szintû protokollok nem lennének képesek.

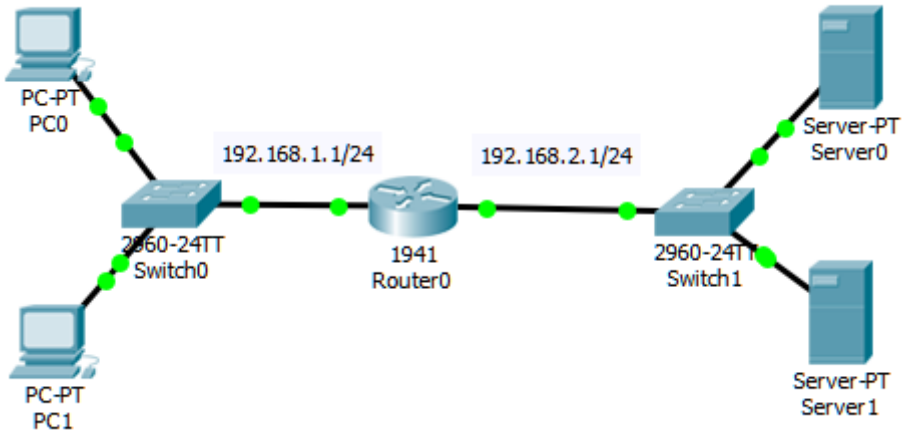
A Frame Relay jelenleg 64 kbit/s és 2 Mbit/s között mûködik, de léteznek kisebb és nagyobb sebességû implementációk. Hamarosan várható a DS3 (45 Mbit/s) vonalakon mûködô Frame Relay berendezések megjelenése. Legyen az magán vagy nyilvános hálózat, az interface Frame Relay volta nem feltétlen jelenti, hogy a hálózat belseje is ezen elvek szerint mûködik. Jelenleg nem léteznek szabványok a Frame Relay hálózatokon belüli berendezések összekapcsolására, így bármilyen technológia felhasználható ilyen szolgáltatás nyújtására.

A Frame Relay szabványosítás alatt álló SVC hívásfelépítési mechanizmusa (Q.933) ugyanarra a filozófiára épül, mint az ISDN (Q.931) és az ATM (Q.2931) jelzésrendszer. Errôl bôvebben az ISDN és ATM fejezetekben lesz szó.

Feladatok

ACL beállítása

Ne lehessen elérni a .200 server



Router(config)#access-list 123 deny ip any host 192.168.2.200
Router(config)#access-list 123 permit ip any any
(deny any)
Router(config)#interf gig0/0
Router(config-if)#ip access-group 123 in

pc ping:
C:\>ping 192.168.2.200

Pinging 192.168.2.200 with 32 bytes of data:

Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.
Reply from 192.168.1.1: Destination host unreachable.

Ping statistics for 192.168.2.200:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>ping 192.168.2.100

Pinging 192.168.2.100 with 32 bytes of data:

Reply from 192.168.2.100: bytes=32 time=1ms TTL=127
Reply from 192.168.2.100: bytes=32 time<1ms TTL=127
Reply from 192.168.2.100: bytes=32 time=10ms TTL=127

Reply from 192.168.2.100: bytes=32 time=10ms TTL=127

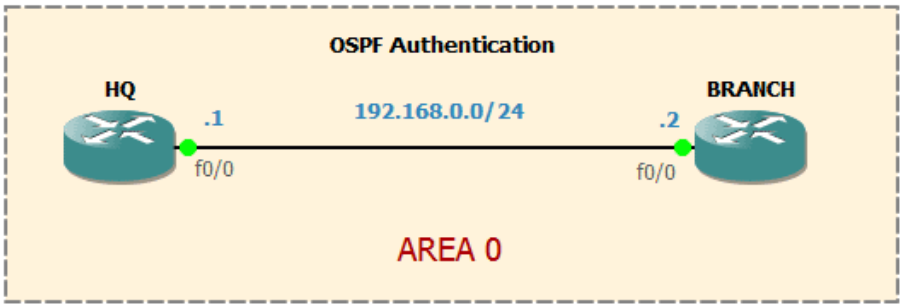
Ping statistics for 192.168.2.100:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

A .10 pc ne pingeljen .100-ra

Router(config)#ip access-list extended no_icmp
Router(config-ext-nacl)#deny icmp host 192.168.1.10 host 192.168.2.100 echo
Router(config-ext-nacl)#permit ip any any

Router(config)#ip access-list extended no_icmp
Router(config-ext-nacl)#deny icmp host 192.168.1.10 host 192.168.2.100 echo
Router(config-ext-nacl)#permit ip any any
Router(config)#ip access-list extended kozos
Router(config-ext-nacl)#deny icmp host 192.168.1.10 host 192.168.2.100 echo
Router(config-ext-nacl)#deny ip any host 192.168.2.200
Router(config-ext-nacl)#permit ip any any
Router(config)#interf gig0/0
Router(config-if)#ip access-group kozos in

OSPF Autentikáció



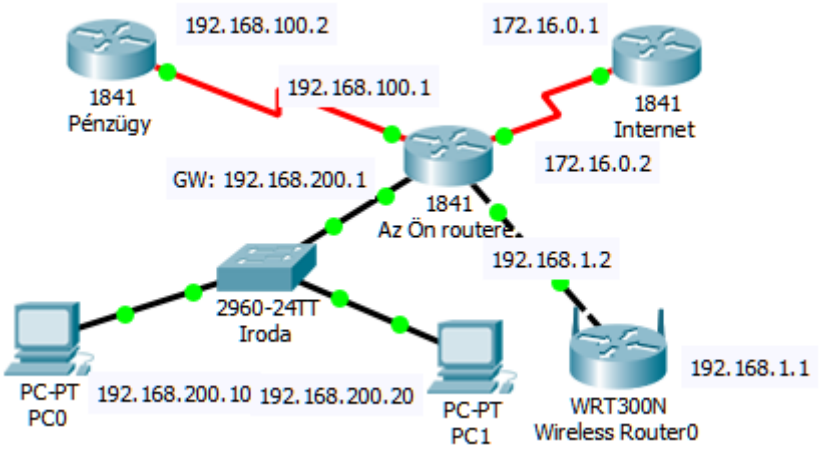

```
HQ(config)#interface fastEthernet 0/0
HQ(config-if)#ip address 192.168.0.1 255.255.255.0
HQ(config-if)#no shutdown

HQ(config)#router ospf 1
HQ(config-router)#network 192.168.0.0 0.0.0.255 area 0

HQ(config)#interface fastEthernet 0/0
HQ(config-if)#ip ospf authentication-key test
HQ(config-if)#ip ospf authentication
```

Ha pedig tiskosítva akarom küldeni a jelyzót:

```
HQ(config)#interface fastEthernet 0/0
HQ(config-if)#ip ospf message-digest-key 1 md5 test
HQ(config-if)#ip ospf authentication message-digest
```

```
BRANCH(config)#interface fastEthernet 0/0
BRANCH(config-if)#ip address 192.168.0.2 255.255.255.0
BRANCH(config-if)#no shutdown

BRANCH(config)#router ospf 1
BRANCH(config-router)#network 192.168.0.0 0.0.0.255 area 0

BRANCH(config)#interface fastEthernet 0/0
BRANCH(config-if)#ip ospf authentication-key test
BRANCH(config-if)#ip ospf authentication

BRANCH(config)#interface fastEthernet 0/0
BRANCH(config-if)#ip ospf message-digest-key 1 md5 test
BRANCH(config-if)#ip ospf authentication message-digest
```

ZPF tűzfal



Zóna alapú tűzfal:

iroda -> penzugy: minden mehet
iroda -> targyalo: semmi sem mehet
iroda -> internet minden mehet kifelé és annak válasza jöhet befelé
pénzugy -> internet minden mehet kifelé és annak válasza jöhet befelé
targyalo -> internet: csak http, icmp, dns és ezek válaszai mehetnek

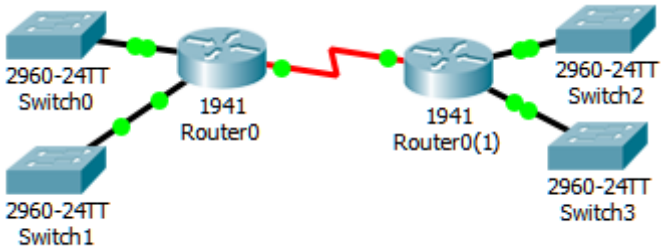
Létrehozom a zónákat:

```
AZONROUTERE(config)#zone security IRODA
AZONROUTERE(config-sec-zone)#exit
AZONROUTERE(config)#zone security PENZUGY
AZONROUTERE(config-sec-zone)#exit
AZONROUTERE(config)#zone security INTERNET
AZONROUTERE(config-sec-zone)#exit
AZONROUTERE(config)#zone security TARGYALO
AZONROUTERE(config-sec-zone)#exit
```

```
AZONROUTERE(config)#

AZONROUTERE(config)#class-map type inspect match-any
MINDENFORGALOM
AZONROUTERE(config-cmap)#match any
AZONROUTERE(config-cmap)#exit
AZONROUTERE(config)#class-map type inspect match-any
WEBFORGALOM
AZONROUTERE(config-cmap)#match protocol icmp
AZONROUTERE(config-cmap)#match protocol http
AZONROUTERE(config-cmap)#match protocol dns
AZONROUTERE(config-cmap)#exit
AZONROUTERE(config)#

AZONROUTERE(config)#policy-map type inspect MINDENMEHET
AZONROUTERE(config-pmap)#class type inspect MINDENFORGALOM
AZONROUTERE(config-pmap-c)#pass
AZONROUTERE(config-pmap-c)#exit
AZONROUTERE(config-pmap)#exit
AZONROUTERE(config)#policy-map type inspect SEMMISEMMEHET
AZONROUTERE(config-pmap)#class type inspect MINDENFORGALOM
AZONROUTERE(config-pmap-c)#drop
AZONROUTERE(config-pmap-c)#exit
AZONROUTERE(config-pmap)#exit
AZONROUTERE(config)#policy-map type inspect CSAKKIFELE
AZONROUTERE(config-pmap)#class type inspect MINDENFORGALOM
AZONROUTERE(config-pmap-c)#inspect
AZONROUTERE(config-pmap-c)#exit
AZONROUTERE(config-pmap)#exit
AZONROUTERE(config)#policy-map type inspect WEBMEHET
AZONROUTERE(config-pmap)#class type inspect WEBFORGALOM
AZONROUTERE(config-pmap-c)#inspect
AZONROUTERE(config-pmap-c)#exit
AZONROUTERE(config-pmap)#exit
AZONROUTERE(config)#

AZONROUTERE(config)#zone-pair security IRODATARGYALO source
IRODA destination TARGYALO
AZONROUTERE(config-sec-zone-pair)#service-policy type inspect
MINDENMEHET
AZONROUTERE(config-sec-zone-pair)#exit
AZONROUTERE(config)#zone-pair security IRODATARGYALO source
IRODA destination TARGYALO
AZONROUTERE(config-sec-zone-pair)#exit
AZONROUTERE(config)#zone-pair security IRODAINTERNET source
IRODA destination INTERNET
AZONROUTERE(config-sec-zone-pair)#service-policy type inspect
CSAKKIFELE
AZONROUTERE(config-sec-zone-pair)#exit
```

```
AZONROUTERE(config)#zone-pair security PENZUGYINTERNET source
PENZUGY destination INTERNET
AZONROUTERE(config-sec-zone-pair)#service-policy type inspect
CSAKKIFELE
AZONROUTERE(config-sec-zone-pair)#exit
AZONROUTERE(config)#zone-pair security PEWNZUGYIRODA source
PENZUGY destination IRODA
AZONROUTERE(config-sec-zone-pair)#service-policy type inspect
MINDENMEHET
AZONROUTERE(config-sec-zone-pair)#exit
AZONROUTERE(config)#zone-pair security TARGYALOINTERNET source
TARGYALO destination INTERNET
AZONROUTERE(config-sec-zone-pair)#service-policy type inspect
WEBMEHET
AZONROUTERE(config-sec-zone-pair)#exit

AZONROUTERE(config)#interf fa0/0
AZONROUTERE(config-if)#zone-member security TARGYALO
AZONROUTERE(config-if)#exit
AZONROUTERE(config)#interf fa0/1
AZONROUTERE(config-if)#zone-member security IRODA
AZONROUTERE(config-if)#exit
AZONROUTERE(config)#interf ser 0/0/1
AZONROUTERE(config-if)#zone-member security INTERNET
AZONROUTERE(config-if)#exit
AZONROUTERE(config)#interf ser 0/0/0
AZONROUTERE(config-if)#zone-member security PENZUGY
AZONROUTERE(config-if)#exit
AZONROUTERE(config)#
```

EIGRP kulcsokkal



Automatikus összevonás

```
R1(config)#router eigrp 50
R1(config-router)#auto-summary
```

Kézi útvonal összevonás

```
R2(config-router)#interf ser0/0/0
```

```
R2(config-if)#ip summary-address eigrp 50 192.168.0.0
255.255.0.0
R2(config-if)#
%DUAL-5-NBRCHANGE: IP-EIGRP 50: Neighbor 195.199.199.197
(Serial0/0/0) is up: new adjacency
```

Kulcs létrehozása

```
R1(config)#key chain eigrp_kulcsok
R1(config-keychain)#key 7
R1(config-keychain-key)#key-string estiosztaly

R1(config)#interf ser0/0/0
R1(config-if)#ip authentication mode eigrp 50 md5
R1(config-if)#ip authentication key-chain eigrp 50
eigrp_kulcsok

R2(config)#key chain kulcsaim
R2(config-keychain)#key 7
R2(config-keychain-key)#key-string estiosztaly
R2(config-keychain-key)#exit
R2(config-keychain)#exit
R2(config)#interf ser0/0/0
R2(config-if)#ip authentication mode eigrp 50 md5
R2(config-if)#ip authentication key-chain eigrp 50 kulcsaim
```

MAC-address szűrés routeren

```
hostname ISKOLA
ip dhcp pool LAN
   network 192.168.88.0 255.255.255.0
   default-router 192.168.88.1
   dns-server 8.8.8.8
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
bridge irb
interface GigabitEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
 bridge-group 1
interface GigabitEthernet0/1
 ip address dhcp
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
interface BRI0/1/0
 no ip address
 encapsulation hdlc
 shutdown
interface GigabitEthernet1/0
 no ip address
 ip nat inside
 ip virtual-reassembly

bridge-group 1
 bridge-group 1 input-address-list 700
interface BVI1
 description BELSO_HALOZAT
 ip address 192.168.88.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly

ip nat inside source list 1 interface GigabitEthernet0/1
overload
!
access-list 1 permit 192.168.88.0 0.0.0.255
access-list 700 permit 001b.384c.ddb6   0000.0000.0000
access-list 700 permit 0021.cc70.ed7c   0000.0000.0000
bridge 1 protocol ieee
bridge 1 route ip
```

LINUX

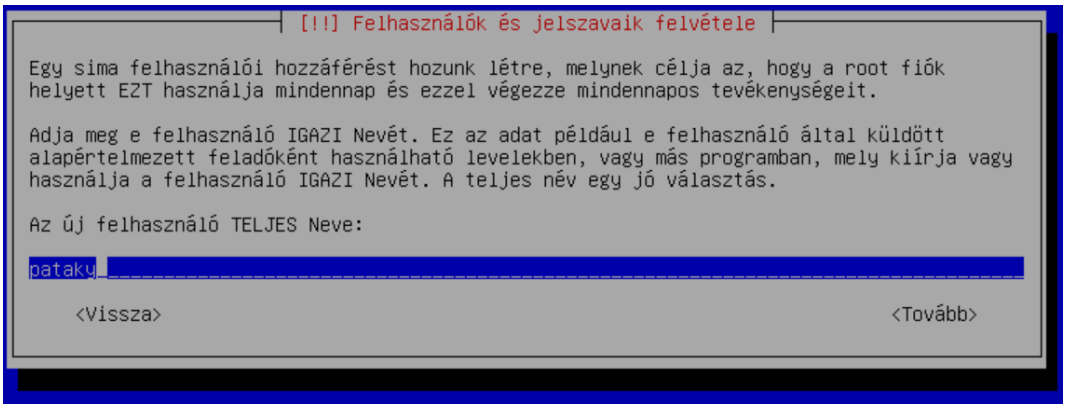
Aszimmetrikus RAID5 telepítése

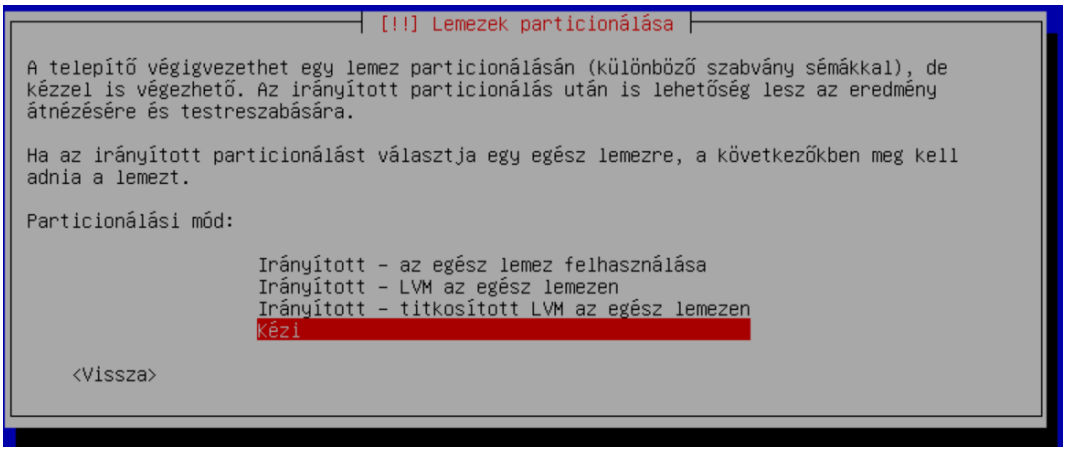
Telepíts egy Debian 9.3.0 (32b netinst.iso) szerver-operációs rendszert az alábbi paraméterekkel:

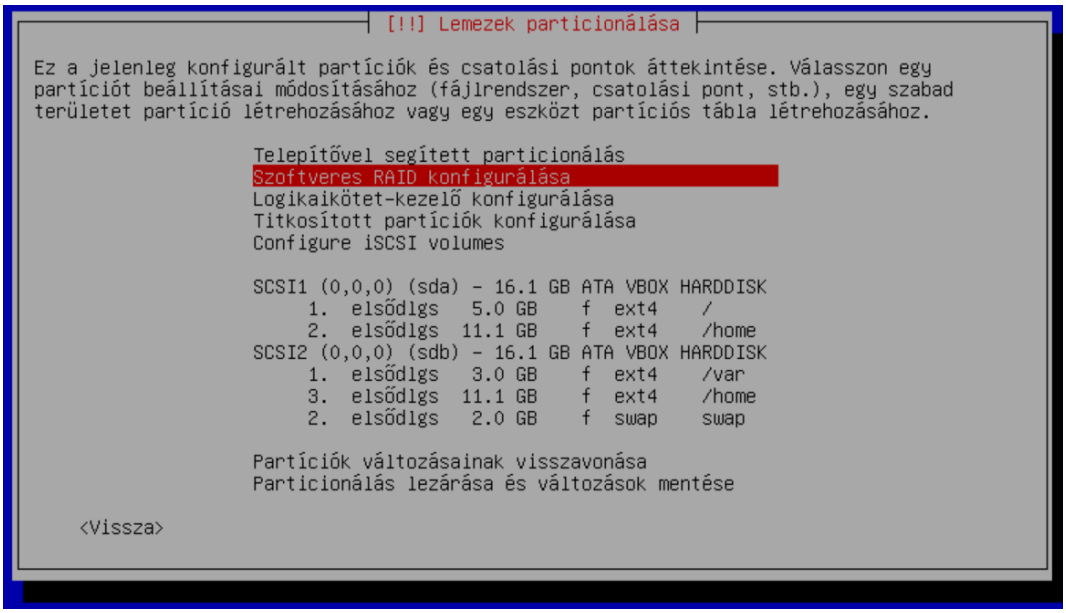
- 2db 15 GB HDD
  - 5 GB / (etx4)
  - 10 GB /home (ext4) **

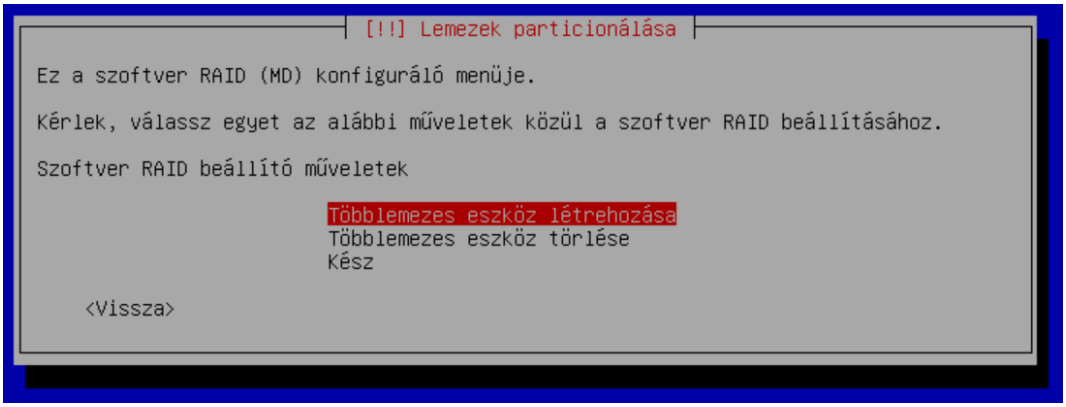
  - 3 GB /var (ext4)
  - 2 GB swap
  - 10 GB /home (ext4) **

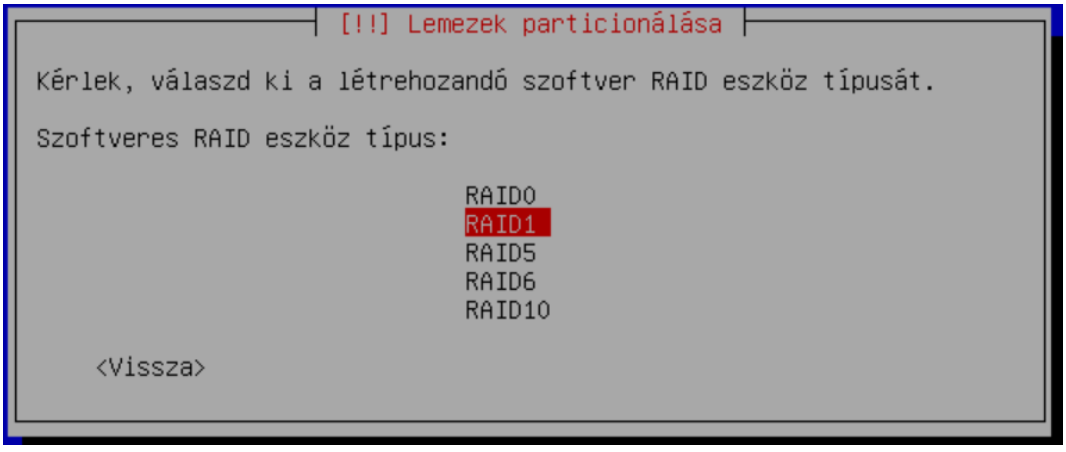
  A két **-al jelölt partíció egymás tükrözése (RAID 1)

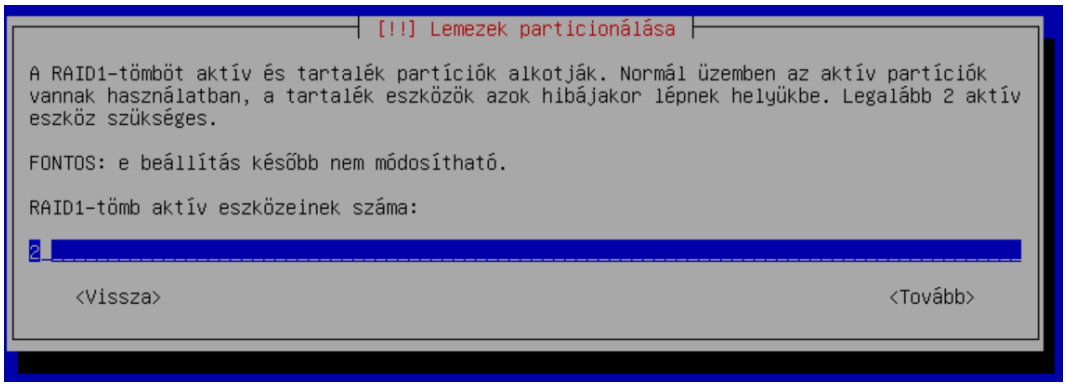


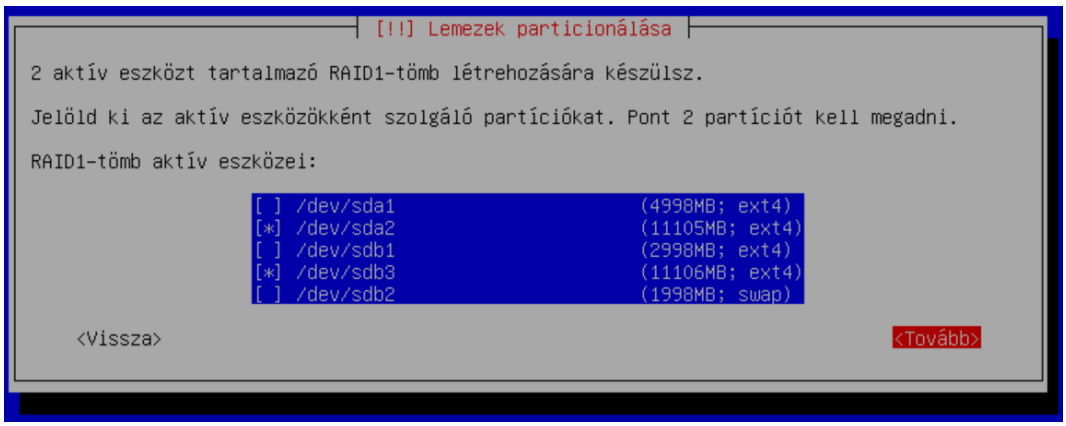


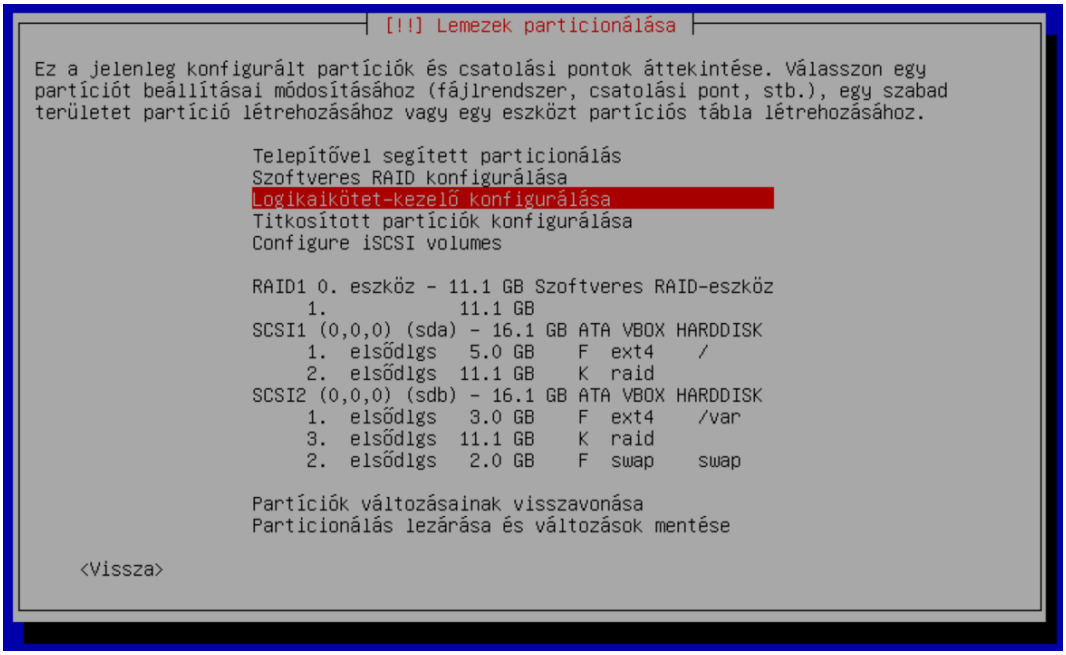

[!!] Lemezek particionálása
Ez a jelenleg konfigurált partíciók és csatolási pontok áttekintése. Válasszon egy partíciót beállításai módosításához (fájlrendszer, csatolási pont, stb.), egy szabad területet partíció létrehozásához vagy egy eszközt partíciós tábla létrehozásához.
Telepítővel segített particionálás
Szoftveres RAID konfigurálása
Logikaikötet-kezelő konfigurálása
Titkosított partíciók konfigurálása
Configure iSCSI volumes
SCSI1 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 5.0 GB f ext4 /
2. elsődlgs 11.1 GB f ext4 /home
SCSI2 (0,0,0) (sdb) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 3.0 GB f ext4 /var
3. elsődlgs 11.1 GB f ext4 /home
2. elsődlgs 2.0 GB f swap swap
Partíciók változásainak visszavonása
Particionálás lezárása és változások mentése
<Vissza>

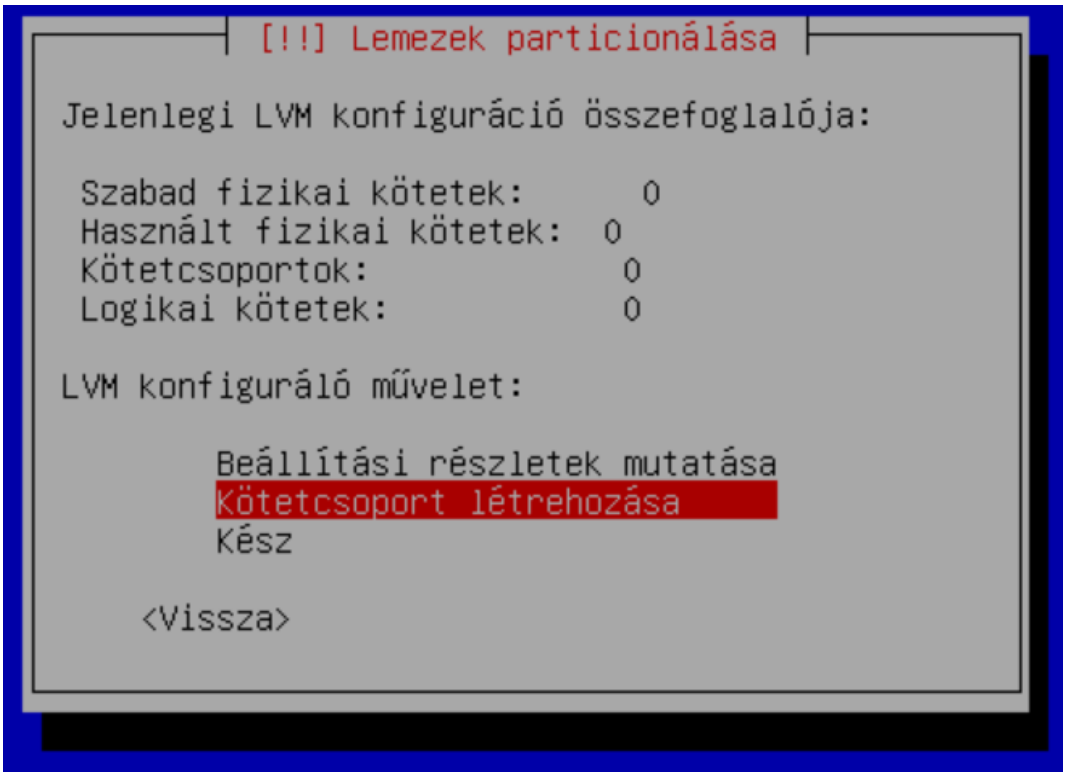

[!!] Lemezek particionálása
Ez a szoftver RAID (MD) konfiguráló menüje.
Kérlek, válassz egyet az alábbi műveletek közül a szoftver RAID beállításához.
Szoftver RAID beállító műveletek
Többlemezes eszköz létrehozása
Többlemezes eszköz törlése
Kész
<Vissza>

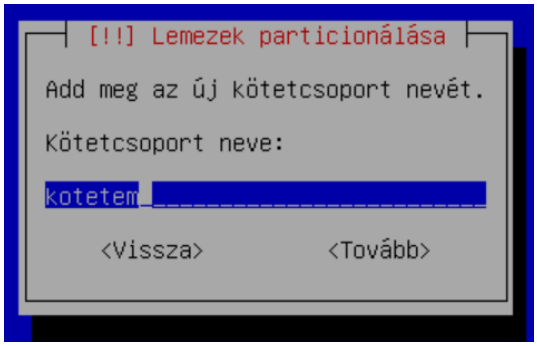

[!!] Lemezek particionálása
Kérlek, válaszd ki a létrehozandó szoftver RAID eszköz típusát.
Szoftveres RAID eszköz típus:
RAID0
RAID1
RAID5
RAID6
RAID10
<Vissza>

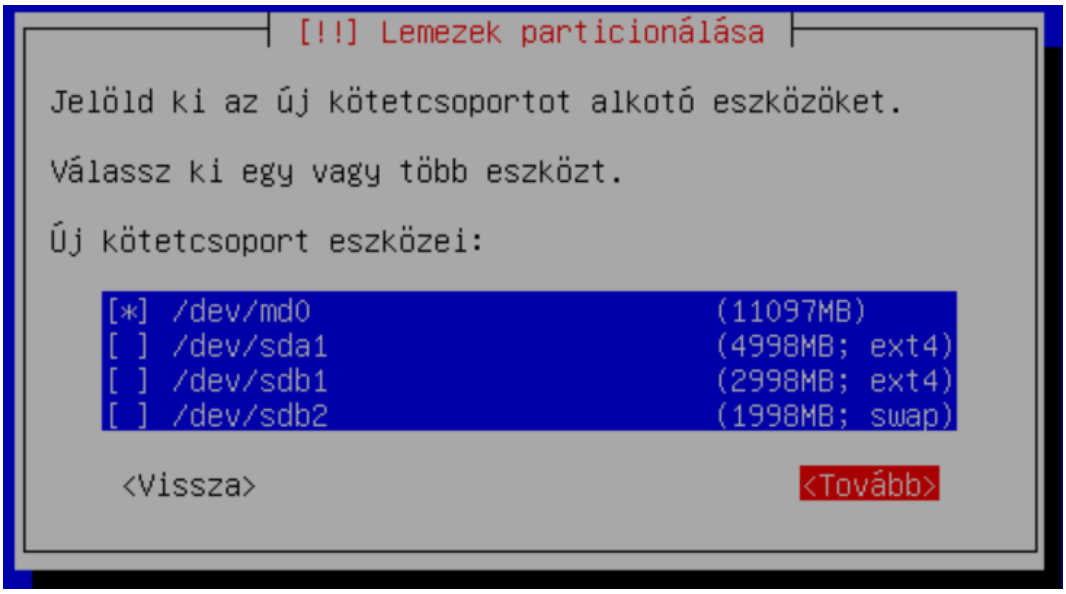

[!!] Lemezek particionálása
A RAID1-tömböt aktív és tartalék partíciók alkotják. Normál üzemben az aktív partíciók vannak használatban, a tartalék eszközök azok hibájakor lépnek helyükbe. Legalább 2 aktív eszköz szükséges.
FONTOS: e beállítás később nem módosítható.
RAID1-tömb aktív eszközeinek száma:
2
<Vissza>
<Tovább>

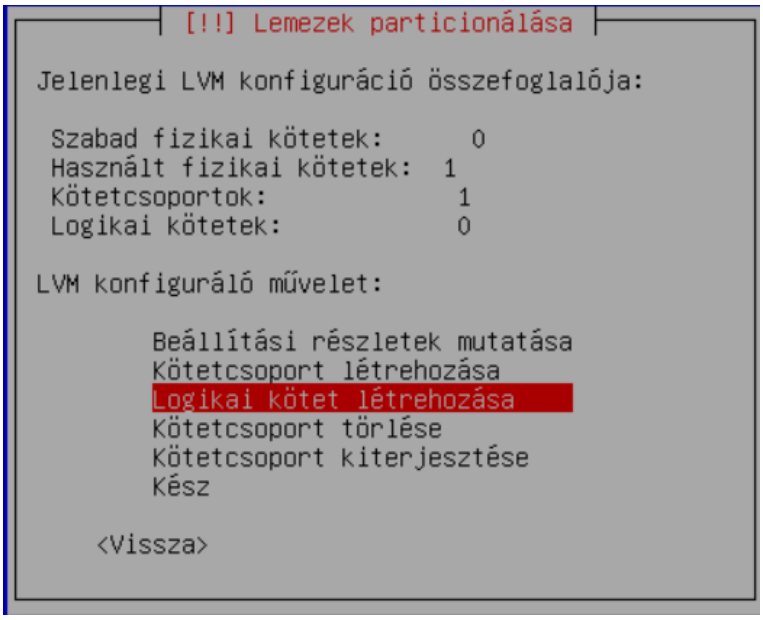

[!!] Lemezek particionálása
2 aktív eszközt tartalmazó RAID1-tömb létrehozására készülsz.
Jelöld ki az aktív eszközökként szolgáló partíciókat. Pont 2 partíciót kell megadni.
RAID1-tömb aktív eszközei:
[ ] /dev/sda1 (4998MB; ext4)
[*] /dev/sda2 (11105MB; ext4)
[ ] /dev/sdb1 (2998MB; ext4)
[*] /dev/sdb3 (11106MB; ext4)
[ ] /dev/sdb2 (1998MB; swap)
<Vissza>
<Tovább>

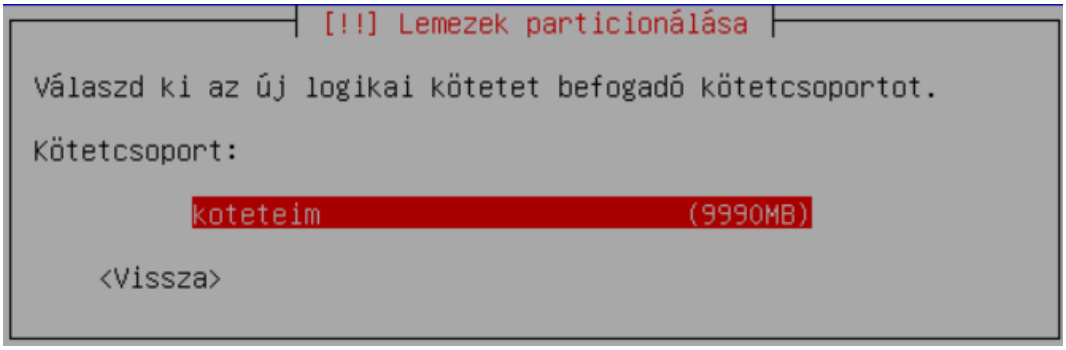

[!!] Lemezek particionálása
Ez a jelenleg konfigurált partíciók és csatolási pontok áttekintése. Válasszon egy partíciót beállításai módosításához (fájlrendszer, csatolási pont, stb.), egy szabad területet partíció létrehozásához vagy egy eszközt partíciós tábla létrehozásához.
Telepítővel segített particionálás
Szoftveres RAID konfigurálása
Logikaikötet-kezelő konfigurálása
Titkosított partíciók konfigurálása
Configure iSCSI volumes
RAID1 0. eszköz - 11.1 GB Szoftveres RAID-eszköz
1. 11.1 GB
SCSI1 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 5.0 GB F ext4 /
2. elsődlgs 11.1 GB K raid
SCSI2 (0,0,0) (sdb) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 3.0 GB F ext4 /var
3. elsődlgs 11.1 GB K raid
2. elsődlgs 2.0 GB F swap swap
Partíciók változásainak visszavonása
Particionálás lezárása és változások mentése
<Vissza>

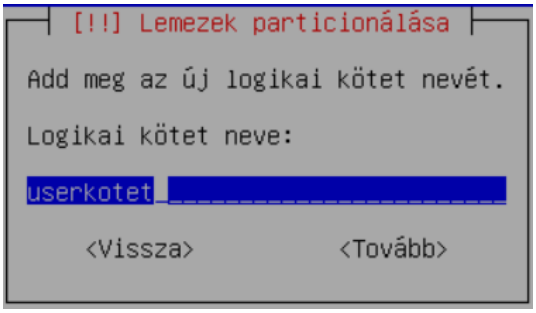

[!!] Lemezek particionálása
Jelenlegi LVM konfiguráció összefoglalója:
Szabad fizikai kötetek: 0
Használt fizikai kötetek: 0
Kötetcsoportok: 0
Logikai kötetek: 0
LVM konfiguráló művelet:
Beállítási részletek mutatása
Kötetcsoport létrehozása
Kész
<Vissza>

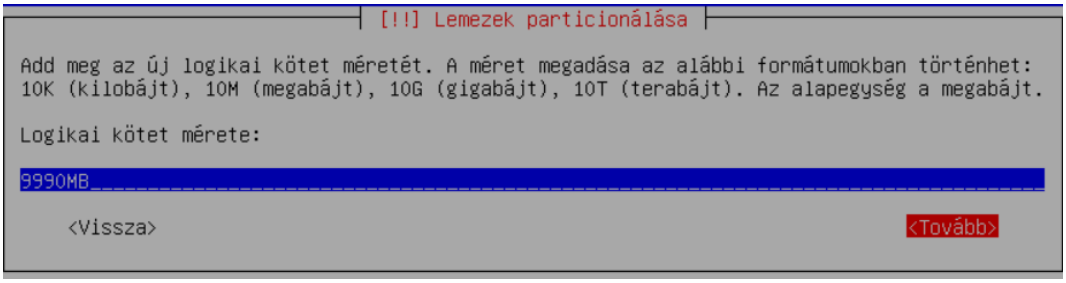

[!!] Lemezek particionálása
Add meg az új kötetcsoport nevét.
Kötetcsoport neve:
kotetem
<Vissza>
<Tovább>

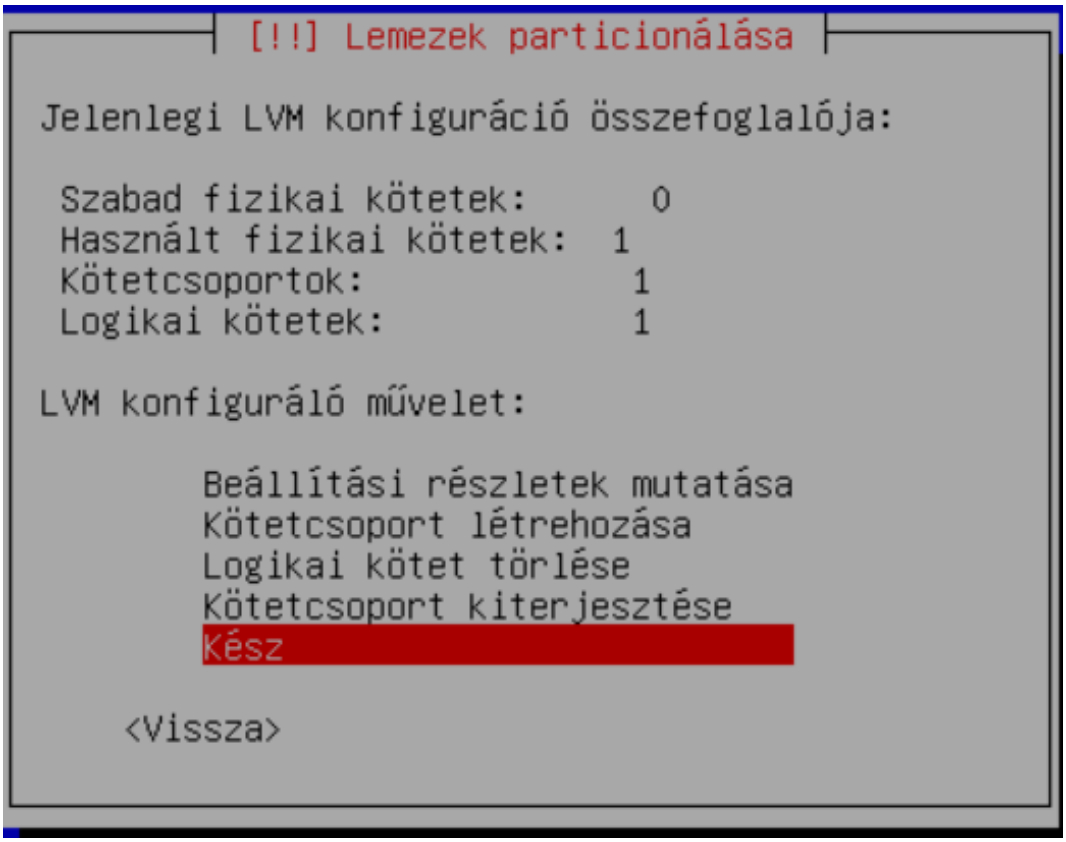

[!!] Lemezek particionálása
Jelöld ki az új kötetcsoportot alkotó eszközöket.
Válassz ki egy vagy több eszközt.
Új kötetcsoport eszközei:
[*] /dev/md0 (11097MB)
[ ] /dev/sda1 (4998MB; ext4)
[ ] /dev/sdb1 (2998MB; ext4)
[ ] /dev/sdb2 (1998MB; swap)
<Vissza>
<Tovább>

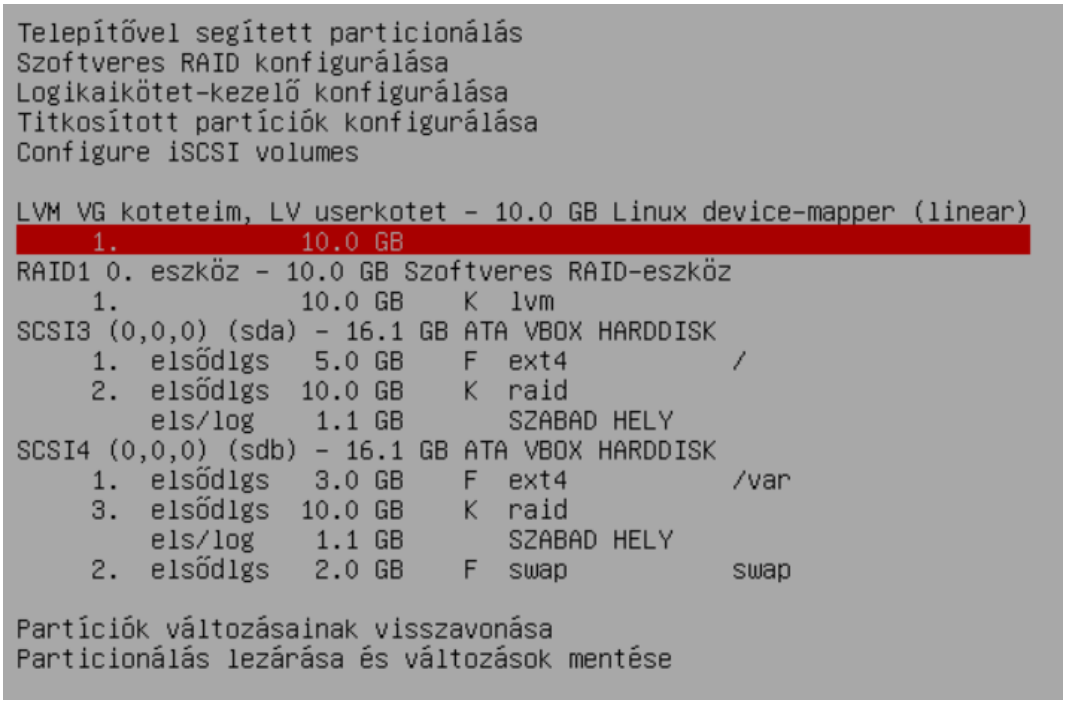

[!!] Lemezek particionálása
Jelenlegi LVM konfiguráció összefoglalója:
Szabad fizikai kötetek: 0
Használt fizikai kötetek: 1
Kötetcsoportok: 1
Logikai kötetek: 0
LVM konfiguráló művelet:
Beállítási részletek mutatása
Kötetcsoport létrehozása
Logikai kötet létrehozása
Kötetcsoport törlése
Kötetcsoport kiterjesztése
Kész
<Vissza>

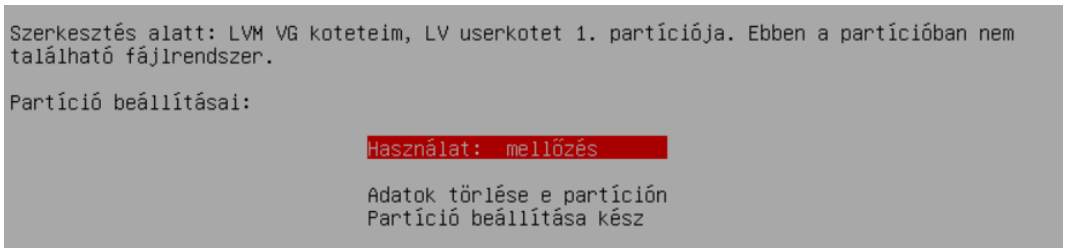

[!!] Lemezek particionálása
Válaszd ki az új logikai kötetet befogadó kötetcsoportot.
Kötetcsoport:
koteteim (9990MB)
<Vissza>

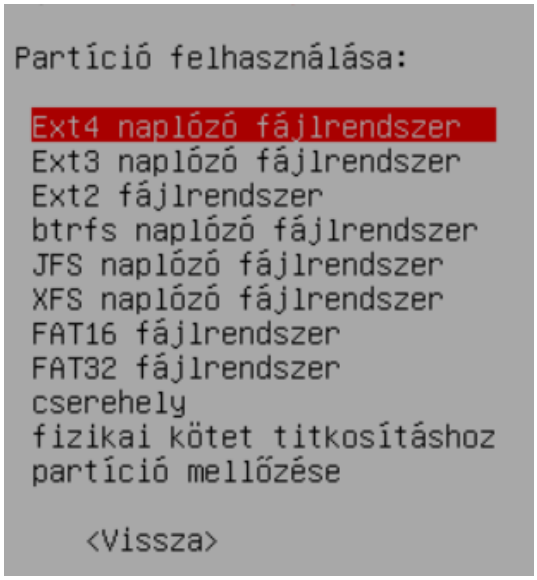

[!!] Lemezek particionálása
Add meg az új logikai kötet nevét.
Logikai kötet neve:
userkotet
<Vissza>
<Tovább>

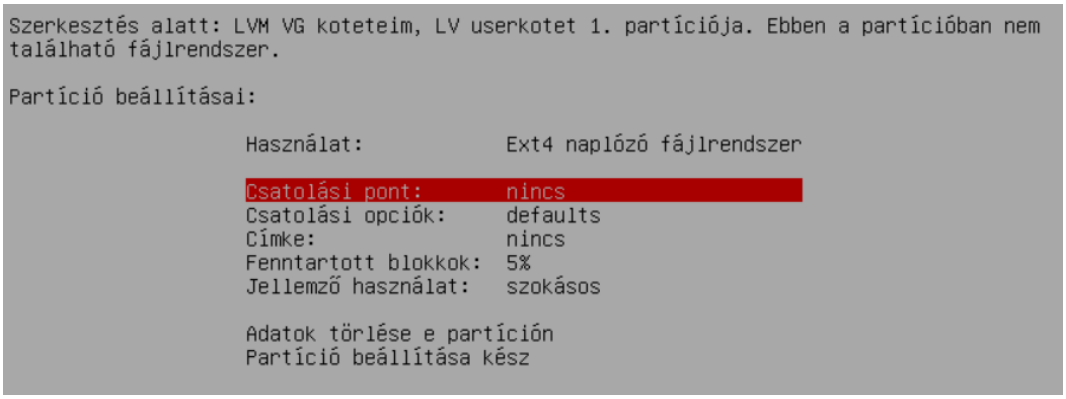

[!!] Lemezek particionálása
Add meg az új logikai kötet méretét. A méret megadása az alábbi formátumokban történhet:
10K (kilobájt), 10M (megabájt), 10G (gigabájt), 10T (terabájt). Az alapegység a megabájt.
Logikai kötet mérete:
9990MB
<Vissza>
<Tovább>

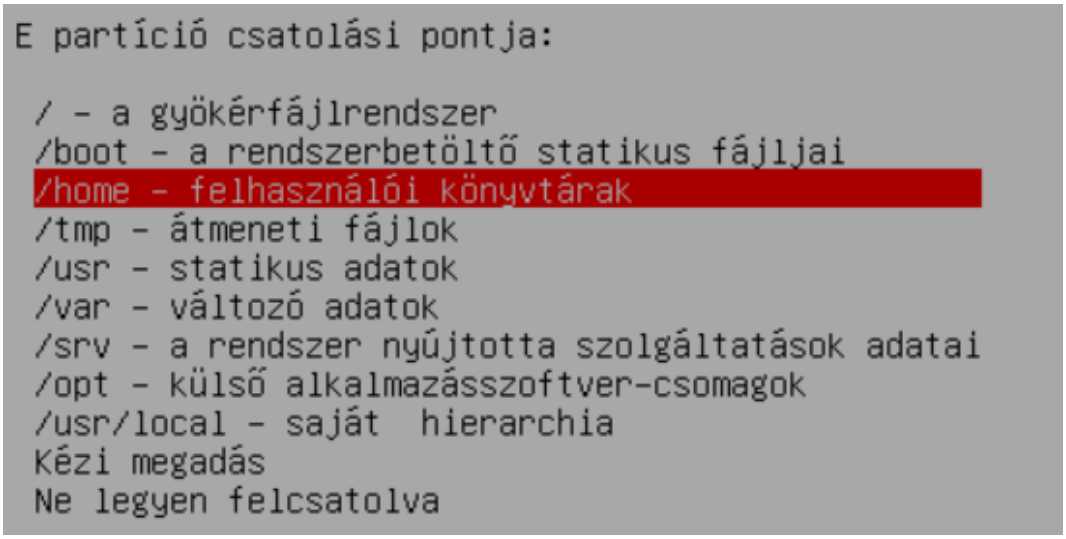

[!!] Lemezek particionálása
Jelenlegi LVM konfiguráció összefoglalója:
Szabad fizikai kötetek: 0
Használt fizikai kötetek: 1
Kötetcsoportok: 1
Logikai kötetek: 1
LVM konfiguráló művelet:
Beállítási részletek mutatása
Kötetcsoport létrehozása
Logikai kötet törlése
Kötetcsoport kiterjesztése
Kész
<Vissza>

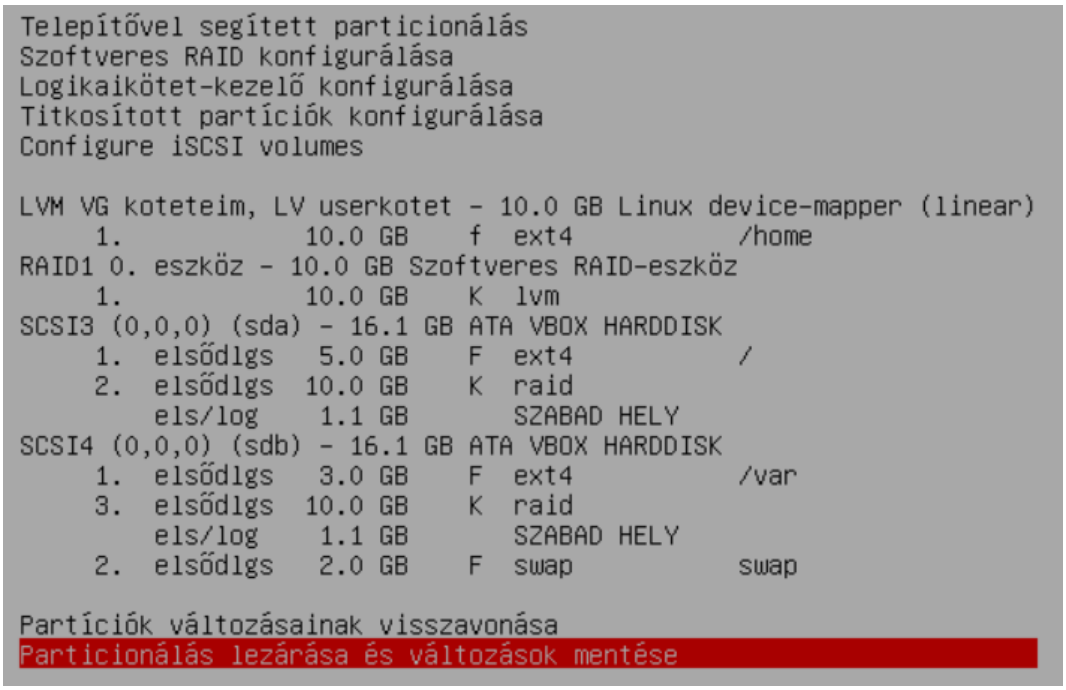

Telepítővel segített particionálás
Szoftveres RAID konfigurálása
Logikaikötet-kezelő konfigurálása
Titkosított partíciók konfigurálása
Configure iSCSI volumes
LVM VG koteteim, LV userkotet - 10.0 GB Linux device-mapper (linear)
1. 10.0 GB
RAID1 0. eszköz - 10.0 GB Szoftveres RAID-eszköz
1. 10.0 GB K lvm
SCSI3 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 5.0 GB F ext4 /
2. elsődlgs 10.0 GB K raid
els/log 1.1 GB SZABAD HELY
SCSI4 (0,0,0) (sdb) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 3.0 GB F ext4 /var
3. elsődlgs 10.0 GB K raid
els/log 1.1 GB SZABAD HELY
2. elsődlgs 2.0 GB F swap swap
Partíciók változásainak visszavonása
Particionálás lezárása és változások mentése


Szerkesztés alatt: LVM VG koteteim, LV userkotet 1. partíciója. Ebben a partícióban nem található fájlrendszer.
Partíció beállításai:
Használat: mellőzés
Adatok törlése e partíción
Partíció beállítása kész


Partíció felhasználása:
Ext4 naplózó fájlrendszer
Ext3 naplózó fájlrendszer
Ext2 fájlrendszer
btrfs naplózó fájlrendszer
JFS naplózó fájlrendszer
XFS naplózó fájlrendszer
FAT16 fájlrendszer
FAT32 fájlrendszer
cserehely
fizikai kötet titkosításhoz
partíció mellőzése
<Vissza>


Szerkesztés alatt: LVM VG koteteim, LV userkotet 1. partíciója. Ebben a partícióban nem található fájlrendszer.
Partíció beállításai:
Használat: Ext4 naplózó fájlrendszer
Csatolási pont: nincs
Csatolási opciók: defaults
Címke: nincs
Fenntartott blokkok: 5%
Jellemző használat: szokásos
Adatok törlése e partíción
Partíció beállítása kész


E partíció csatolási pontja:
/ - a gyökérfájlrendszer
/boot - a rendszerbetöltő statikus fájljai
/home - felhasználói könyvtárak
/tmp - átmeneti fájlok
/usr - statikus adatok
/var - változó adatok
/srv - a rendszer nyújtotta szolgáltatások adatai
/opt - külső alkalmazásszoftver-csomagok
/usr/local - saját hierarchia
Kézi megadás
Ne legyen felcsatolva


Telepítővel segített particionálás
Szoftveres RAID konfigurálása
Logikaikötet-kezelő konfigurálása
Titkosított partíciók konfigurálása
Configure iSCSI volumes
LVM VG koteteim, LV userkotet - 10.0 GB Linux device-mapper (linear)
1. 10.0 GB f ext4 /home
RAID1 0. eszköz - 10.0 GB Szoftveres RAID-eszköz
1. 10.0 GB K lvm
SCSI3 (0,0,0) (sda) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 5.0 GB F ext4 /
2. elsődlgs 10.0 GB K raid
els/log 1.1 GB SZABAD HELY
SCSI4 (0,0,0) (sdb) - 16.1 GB ATA VBOX HARDDISK
1. elsődlgs 3.0 GB F ext4 /var
3. elsődlgs 10.0 GB K raid
els/log 1.1 GB SZABAD HELY
2. elsődlgs 2.0 GB F swap swap
Partíciók változásainak visszavonása
Particionálás lezárása és változások mentése

Banner – belépési üzenet

 # nano /etc/motd

SSH server telepítése és beállítása

 # apt-get install openssh-server

 # nano /etc/ssh/sshd_config (/etc/ssh/ssh_config ->ssh client)

```
Port 22222
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::
```

 # cat /etc/ssh/sshd_config | grep Port – az ellenőrzéshez

Hogy Root ne léphessen b e ssh-val:

```
#LoginGraceTime 2m
PermitRootLogin no
```

```
login as: root
root@192.168.0.32's password:
Access denied
```

 # /etc/init.d/ssh stop | start | restart

 # apt-get purge openssh-server ha valami balul sülne el.

$ ssh remote_username@remote_host a kapcsolódáshoz

Vagy

apt-get install ssh

beállítófájlja: **/etc/ssh/ssh_config** ->
szolgáltatásinduláskor van hatása

Fontosabb részek:
port 22 -> port VALAMIMÁS
permitrootlogin yes -> permitrootlogin no

Szolgáltatás leállítása: **service ssh stop** VAGY
/etc/init.d/ssh stop

Szolgáltatás indítása: **service ssh start** VAGY
/etc/init.d/ssh start

Szolgáltatás újraindítása: **service ssh restart** VAGY
/etc/init.d/ssh restart

A szolgáltatás leállításával a már élő kapcsolatok nem szakadnak meg.

FTP server telepítése

 # apt-get install proftpd
 # nano /etc/proftpd/proftpd.conf
Itt állítom be a port számát és az Anonymus hozzáféréseket alapesetben tiltott a root-tal és a névtelenül belépés egyéb felhasználóval beléphetünk, belépés után NEM a felhasználó saját mappája (/home/username) látszik, hanem a root mappa.

Karakteres felületről az **ftp** paranccsal, vagy az **mc**-nek a beépített ftp kliensével tudunk ftp szerverhez csatlakozni: **ftp user@szervermnév** vagy **ftp user@ipcím.**

A beállítófájl a **/etc/proftpd** mappában vannak. A fő beállítások a **proftpd.conf**-ban vannak.

Ahhoz, hogy bejelentkezés után a saját mappáját lássa a felhasználó, a DefaultRoot **~** opció elől vegyük ki a kommentezést. A szerver újraindítása után lép érvénybe a változtatás. (A **~** a felhasználó saját mappáját jelenti – pl. **cd ~ ?** belép a saját mappába.)

A névtelen bejelentkezés lehetővé tételéhez a konfigállományban az **<Anonymous ~ftp>** és **</Anonymous>** közötti sorok elől egy db # jelet töröljünk, és indítsuk újra a szolgáltatást.

Ebben az esetben a név nélkül bejelentkezők a **/srv/ftp** mappába kerülnek a bejelentkezés után.

 # /etc/init.d/proftpd stop | start | restart
 # apt-get purge proftpd-basic

FTP részletes beállításai

ServerName {string}

A kiszolgáló nevét határozza meg. A kapcsolódást követõen alapértelmezésben ezt a szövegfüzért látják a parancssori ftp-t használók:

```
"ProFTPD 1.2.5rc1 Server (Debian)
[inter.net]"
```

Itt a Debian a ServerName által meghatározott név, míg az inter.net a számítógép úgynevezett FQDN-je (Fully

Qualified Domain Name), vagyis a tartománynév.

• ServerType {"inetd" | "standalone"}

Itt a kiszolgálófolyamat futtatásának módját határozod meg. Az inetd, illetve standalone közötti különbség

leírását lásd fentebb.

• User {string}

Annak a felhasználónak a neve, aki a démon tulajdonosa. Debian alatt alapértelmezésben ez nobody, viszont ezt a felhasználót más folyamatok is használják, így érdemes

egy külön ftpd nevû felhasználót létrehozni /bin/false héjjal, és azt adni meg itt.

• Group {string}

Annak a csoportnak a neve, aki a démon tulajdonosa. Szintén ajánlott a nogroup helyett egy ftpd nevû felhasználó létrehozása.

• DefaultRoot {string} [string]

Ez a parancs az angol szakszókincs szerint egy jail root-ot hoz létre, ami valójában egy ketrec a felhasználóidnak. Az elsõ értékként megadott könyvtárból nem tudnak feljebb lépni, vagyis számukra ez az új gyökér. A könyvtárnév az egyes felhasználókra nézve lehet relatív. A második érték nem kötelezõ, ezzel egy vagy több csoportra engedélyezheted avagy tilthatod ezt a megszorítást. Például:

"DefaultRoot ~ download,!upload"

Ennek a segítségével eléred, hogy minden felhasználó, aki tagja a download csoportnak, de nem tagja az upload csoportnak, be legyen szorítva a saját könyvtárába (~).

• AllowForeignAddress {"on" | "off"}

Engedélyezi az ügyfélnek, hogy a PORT ftp parancs használatakor a sajátján kívül más címét is használhassa.

Ha nem érted, hogy ez mit jelent, akkor nem olvastad el az rfc-t, erre viszont most nem térnék ki. A lényeg, hogy ha ezt bekapcsolod (alapértelmezésben tiltva van), akkor a felhasználók igénybe vehetik az FXP nyújtotta lehetõségeket.

Az FXP egy olyan módszer (nem külön protokoll), amellyel két kiszolgáló között anélkül mozgathatsz állományokat, hogy a saját gépedre letöltenéd õket.

• AuthUserFile {string}

Lehetõség van a proftpd-ben más forrásokból is meríteni a felhasználói adatbázist, nem kötelezõ a /etc/passwd-t használni. Ez azért remek, mert az FTP-felhasználóknak nem kell feltétlenül létezniük a rendszerben. E források között szerepel az LDAP-, az SQL-kiszolgálók, illetve egy másik passwd állomány. Ezt az utóbbi forrást használja ez az irányelv, ami már elég nagyfokú biztonságot tesz lehetõvé.

Gondolj csak arra, hogy így könnyen megakadályozhatod az FTP-felhasználók távoli belépését telnet-en vagy ssh-n keresztül. Mivel azonban itt nem létezik az árnyékjelszó fogalma, óvatosan állítsd be a jogosultságokat arra az állományra, amit itt értékként megadsz. A legjobb, ha a tulajdonosa

és a csoportja az, amit a User, illetve a Group meghatározásnál megadtál, és csak a tulajdonos és a csoport

tudja írni és olvasni, a többieknek pedig semmilyen jogosultságuk nincs. Az állományt az ftpasswd parancs segítségével lehet létrehozni, illetve karbantartani, közvetlen írása nem ajánlott.

• AuthGroupFile {string}

A hasonló nevû AuthUserFile-hoz hasonló, csak ez a /etc/group állományt helyettesíti. Szintén az ftpasswd

paranccsal illik írni.

Az ftpasswd

Ez egy olyan könnyen használható segédeszköz, mely az AuthUserFile vagy az AuthGroupFile által megadott

állományt módosítja. Kapcsolóit két egyszerû példán keresztül mutatom be. Létrehozok egy andras nevû felhasználót, és a download nevû csoportba teszem bele.

ftpasswd --passwd --name andras --uid 1000

--gid 100 --home /home/andras --shell /bin/sh

A --passwd határozza meg, hogy a felhasználók adatbázisát szeretném módosítani, nem pedig a csoportokét. Lehetõség nyílik egy --file kapcsoló használatára, így meg lehet határozni az adatbázis nevét. Ha elhagyod, alapértelmezés szerint ./ftpd.passwd, azaz a pillanatnyi könyvtárban egy ftpd.passwd nevû állomány. A többi kapcsoló magától értetõdõ. A --gid-et elhagyhatod, ekkor egy a uid-del megegyezõ csoportazonosítót

vesz alapul. Senkit ne tévesszen meg, hogy létezik egy --shell kapcsoló, és egy valós héj van megadva utána!

Ez nem jelenti azt, hogy az adott felhasználó be tudna lépni távolról telnet-en vagy ssh-n, hiszen nem is létezik a rendszerben. Mindössze a PAM (Pluggable Authentication Modules)

miatt van itt szükség egy valós héj megadására. Többek közt így lehet egy FTP-felhasználót „hibernálni”: /bin/false-t adsz meg héjnak, legyen szó akár AuthUserFile-ról, akár nem. A felhasználó jelszavát ezután kétszer egymás után kell begépelned, akárcsak a passwd parancsnál. Ez a jelszó az

adatbázisban titkosítva tárolódik.

ftpasswd --group --name download --gid 100

A --group azt mondja meg, hogy a csoportok adatbázisát módosítom. A --file elhagyása miatt az állomány neve

./ftpd.group. Paranoiások figyelmébe ajánlom a --enablegroup-passwd kapcsolót, ami szerintem ebben az esetben teljesen felesleges.

A Netfilter beállítása

FTP-kiszolgálót tûzfallal ellátni eléggé összetett feladat. A Netfilter ip_conntrack_ftp modulja ugyanakkor jelentõsen leegyszerûsíti ezt a feladatot. Ez nyomon követi az ftp csomagokat, így azt sem szükséges tudnod, mi az az ftp-data kapu. Ha van ilyen modulod, egy szempillantás alatt beállítható (lásd a 45. CD Magazin/proftpd könyvtárában).

Ha a tûzfaladnak nincs ftp-nyomkövetõ modulja, egy kicsit nehezebb lesz a dolgod. Elõször is engedélyezned kell az ftp (21), az ftp-data (20) kapukat, illetve a passzív átvitelhez mindazokat a kapukat, amik szóba jöhetnek. Ez érthetõ, hiszen az ügyfélnek el kell tudna érni azt a kaput, ahol az átvitel

folyik. A gond csak az, hogy a kapu az 1024-65 535 tartomány bármelyik eleme lehet. Ha ezt mind engedélyezni akarod, ne is telepíts tûzfalat a kiszolgálóra. Azzal lehet segíteni az ügyön, hogy megmondod a proftpd-nek, hogy egy szûk tartományból válasszon magának passzív kaput, és csak azt engedélyezed

a tûzfalban. Az utóbbi varázslat a PassivePorts meghatározással hozható létre.

PassivePorts {int} {int}

Az elsõ egész szám az intervallum alsó, míg a második a felsõ határát jelöli (közöttük csak szóköz van). Fontos, hogy ez a tartomány elég nagy legyen ahhoz, hogy az elvárt számú kapcsolatot ki tudja elégíteni. Ennek megfelelõen egy ip_conntrack_ftp-t nem használó IP Tables tûzfalas FTPkiszolgáló a 2. listán látható módon nézhet ki (lásd a 45. CD Magazin/proftpd könyvtárában).

 # apt-get install apache2
 # apt-get install mcrypt – a PHP-hoz
 # apt-get install phpmyadmin – az SQL adatbázishoz
 # dpkg-reconfigure phpmyadmin – ha újra kell konfigurálni a beállításokat
 # /etc/init.d/apache2 stop | start | restart – minden változtatás után

Többféle érhető el. Statikus oldalakra szakosodott, vagy komplexebb webkiszolgálók. Alapértelmezésben az apache2 települ (telepítőben wekiszolgálót kérve).

Kézi telepítése: **apt-get install apache2**

Futás ellenőrzése: **ps aux |grep apache2**

Szolgáltatás ellenőrzése: **nmap localhost –p 80**

Beállítása: **/etc/apache2/***

Működése: moduláris. A telepített modulok a **/etc/apache2/mods-available**, ezek közül azok töltődnek be, amikre van symlink a /etc/apache2/mods-enabled –ben.

Ez a kiszolgáló több weboldal kiszolgálását képes ellátni, ezeknek az „engedélyezése” is hasonló struktúrában valósul meg (**/etc/apache2/sites-available**, **/etc/apache2/sites-enabled**).

Az alapértelmezetten kiszolgált weblap helye: **/var/www/html** , és ha nincs megadva a kért URL-ben dokumentum, akkor az **index.html** szolgálódik ki. (Ha php vagy más extend modul is telepítve van, akkor index.php vagy index.aspx, stb.)

php futtatómodul hozzáadása apache2-höz

Ahhoz, hogy a .php fájlokban megadott php kódok végrehajtódjanak, szükséges egy php modul telepítése az apache2-höz.

apt-cache search php | grep apache

apt-get install libapache2-mod-php5

Ez után a .php fájlokat lekérve az azokban lévő php kód működni kezd.

mysql (kliens) kiegészítő telepítése php-hoz (és így apache –hoz)

Ahhoz, hogy a php kódokban a mysql-hez kötődő parancsok hiba nélkül lefussanak, elengedhetetlen a mysql modul a php-ben.

apt-cache search mysql |grep php

apt-get install php5-mysql

Újraindítani nem kell semmit, ellenőrizni egy php fájlban elhelyezett <?php phpinfo(); ?> függvényt.

mySQL szerver telepítése, alapvető beállítása

A mysql kiszolgáló igényel egy root mysql-felhasználót, ami nem egyezik meg a rendszer root felhasználóval (jelszava lehet, sőt, legyen más).

Telepítés:

apt-cache search mysql |grep server

apt-get install mysql-server

Ellenőrzése: **nmap localhost ?** erre a **3306** –nak nyitva kell lennie. Azon keresztül fogad külső kéréseket a mysql szerver, DE belső kérések normálisan nem erre jönnek, hanem egy socket-re.

A mysql felhasználók listája ÜRES.

A mysql (nem beépített) adatbázisok listája ÜRES.

Ezek kezelése lehetséges a **mysql-client** –ben parancssorral (**mysql** parancs), vagy pl. **phpmysql** segítségével (webes felület), stb.

CLI kapcsolódás: **mysql –u root -p**, majd benne lehet futtatni a mysql lekérdezésket.

Adatok másolása CLI-n egérrel

Ehhez szükség van egy **kiegészítő daemonra**, ez a **gpm**.

Ha telepítve van, akkor az egérrel kattint húz művelet kijelöl, a jobbkliekk a kijelölt részt beilleszti a kurzorhoz.

DHCP server - uDHCPd

Csak egy interfészen tud dhcp-t szolgáltatni

 # apt-get install udhcpd

Hogy lehessen átjáróként használi a szervert:

```
# echo "1">/proc/sys/net/ipv4/ip_forward
# iptables -t nat -I POSTROUTING -o KÜLSŐINTERFÉSZ -j MASQUERADE
```

 # nano /etc/udhcpd.conf -> beállítom a paramétereket

nano /etc/udhcpd.conf paranccsal szerkesszük a feltelepített szolgáltatás konfigurációs állományát, ahol elsőnek az állomány elején a kioszthető címtartomány határaid állítsuk be, illetve az interfész nevét, amelyikre a szolgáltatást be akarjuk állítani

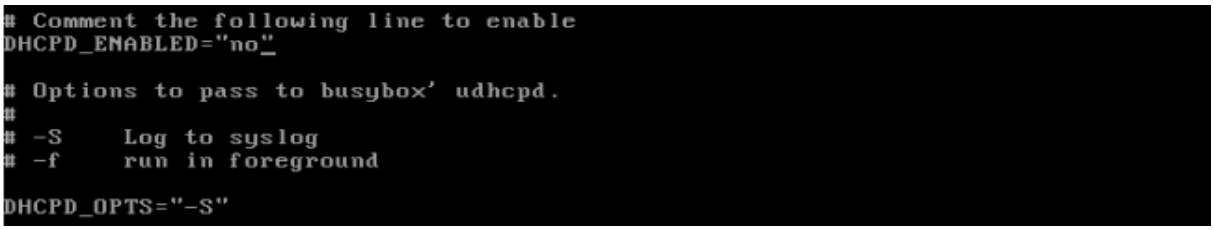
```
# The start and end of the IP lease block

start           192.168.70.20   #default: 192.168.0.20
end             192.168.70.25   #default: 192.168.0.254


# The interface that udhcpd will use

interface       enp0s8          #default: eth0
```

A konfigurációs állomány vége fele található #Examples blokkban a kijelölt sorok szerkesztésével állítsunk be DNS-t (a jó öreg Google-t), maszkot (konfigba router kulcsszó után) és átjárót (konfigba router kulcsszó után). At átjáróhoz az interfészünk fix IP címtét adjuk meg, a blokk többi sorát kikommenteljhetjük:

```
#Examles
opt     dns     8.8.8.8
option  subnet  255.255.255.0
opt     router  192.168.70.1
#opt    wins    192.168.70.1
#option dns     129.219.13.81   # appened to above DNS servers for a total of 3
#option domain  local
#option lease   864000          # 10 days of seconds
```

 # nano /etc/default/udhcp -> dhcp enabled YES

Mielőtt elindítjuk a szolgáltatást, még egy helyen engedélyezni kell a címosztást, ehhez a nano /etc/default/udhcp paranccsal szerkesztenünk kell ezt a konfog fájlt is, ahol a Debian-ban alapból le van tiltva a címszórás! A 2.sorban a „no” kifejezést „yes”-re módosítandó:



 # ps aux | grep dhcp -> fut-e a dhcp szolgáltatás

isc-dhcp-server

 # apt-get install isc-dhcp-server
 # nano /etc/dhcp/dhcpd.conf

Hálózat beállítása

```
    subnet 192.168.3.0 netmask 255.255.255.0 {
        range 192.168.3.10 192.168.3.45;
        range 192.168.3.55 192.168.3.60;
        option routers 192.168.3.1;
        option broadcast-address 192.168.3.255;
        default-lease-time 3600;
        max-lease-time 7200;
    }

        option domain-name-servers 192.168.3.51 192.168.3.54;
        option domain-name "neunetworks.com";
```

A következő parancs arra szolgál, hogy a DHCP-kiszolgáló hiteles szerver legyen a helyi hálózat számára.

authoritative;

A következő parancs a dhcpd.conf fájl szintaxisának ellenőrzésére szolgál hiba esetén

dhcpd -t

A következő parancs lehetővé teszi az isc-dhcp-server fájl szerkesztését, ahol manuálisan meghatározhatjuk a kiszolgáló felületét.

 # nano /etc/default/isc-dhcp-server

Be kell állítanunk az interfészeket a **enp0s8** fájlba

INTERFACES= "enp0s8"

A következő sor jeleníti meg az aktuális útválasztási táblák tartalmát.

ip route

A DHCP szerver biztosítása A konfigurációs fájlhoz a DHCP szerver biztonságának biztosítása érdekében a következő módosítások folynak

```
ddns-update-style none;
deny declines;
deny bootp;
```

A DHS-kiszolgálóval szembeni DoS-támadás elkerülhető a DHCP visszautasító üzenetek megtagadásával, és tagadhatja a régi bootp ügyfelek támogatását.

Majd újra kell indítani a szolgáltatást

 # /etc/init.d/isc-dhcp-server restart | stop | start

ellenőrzése

```
cd /var/lib/dhcp/
ls –l
cat dhcpd.leases
```

Hálókártya beállítása

 # apt-get install net-tools
$ ifconfig -> itt látom a már meglévő beállításokat
$ ifconfig vagy **route -n**

```
root@debvirtbox:/home/pataky# ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.32  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::a00:27ff:fe79:2692  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:79:26:92  txqueuelen 1000  (Ethernet)
        RX packets 266  bytes 296153 (289.2 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 92  bytes 7585 (7.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

$ dmesg | grep eth

```
root@debvirtbox:/home/pataky# dmesg | grep eth
[    7.067683] e1000 0000:00:03.0 eth0: (PCI:33MHz:32-bit) 08:00:27:79:26:92
[    7.077726] e1000 0000:00:03.0 eth0: Intel(R) PRO/1000 Network Connection
[    8.024434] e1000 0000:00:08.0 eth1: (PCI:33MHz:32-bit) 08:00:27:2f:06:90
[    8.024539] e1000 0000:00:08.0 eth1: Intel(R) PRO/1000 Network Connection
[    8.064689] e1000 0000:00:08.0 enp0s8: renamed from eth1
[    8.074926] e1000 0000:00:03.0 enp0s3: renamed from eth0
root@debvirtbox:/home/pataky#
```

vagy
$ lspci (lsusb)

```
00:03.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controll
er (rev 02)
00:04.0 System peripheral: InnoTek Systemberatung GmbH VirtualBox Guest Service
00:06.0 USB controller: Apple Inc. KeyLargo/Intrepid USB
00:07.0 Bridge: Intel Corporation 82371AB/EB/MB PIIX4 ACPI (rev 08)
00:08.0 Ethernet controller: Intel Corporation 82540EM Gigabit Ethernet Controll
```

FONTOS!!! A fájl átírását csak az átállítani kívánt kártya leállítása után végezzük el! (Különben a kernel „eltéved” a beállításokban… -> szerver (!) újraindítás)

 # ifdown KÁRTYANEVE(enp0S3) -> lekapcsolás
 # ifup KÁRTYANEVE(enp0S3) -> felkapcsolás

 # nano /etc/network/interfaces szerkesztésével módosítom a beállításokat
Az interfaces fájlban minden kártyához tartozik egy-egy szekció. Ebben lehet egy auto ethN sor -> azt adja meg, hogy automatikusan felkapcsolandó-e az interfész a rendszerinduláskor.

Továbbá kötelezően van egy iface ethN inet **MODE** sor is, ahol a MODE lehet **dhcp** – ekkor DHCP kliensként működik az a kártya, **manual** – ekkor nem állítódik be a kártya sehogy, **static** – ekkor a következő sorokban megadott fixip használandó.

A kártyához tartozó további sorokat illik bentebb kezdeni (tab), és tartalmuk az alábbiak lehetnek:

address A.B.C.D -ez a kártya címe
netmask E.F.G.H -ez az alhálómaszk
gateway I.J.K.L -ez az alapért. átjáró

továbbá opcionálisan:

broadcast M.N.O.P -a szórási cím
network Q.R.S.T -a hálózat címe

és ezek sorrendje tetszőleges.

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

# DHCP-re beallitott halokartya
auto enp0S8
iface enp0S8 inet static
        address 192.168.0.250
        netmask 255.255.255.0
        gateway 192.168.0.1
```

Hálókártya beállítása NMCLI segítségével

Az nmcli szintaxisa:

```
  # nmcli [OPTIONS] OBJECT {COMMAND |  help}
```

Ahol az OBJECT egyike: általános, hálózati, rádió, kapcsolat, eszköz, ügynök.

Jó kiindulópont lenne a készülékeink ellenőrzéséhez:

```
# nmcli dev status
DEVICE      TYPE      STATE         CONNECTION
docker0     bridge    connected     docker0
virbr0      bridge    connected     virbr0
enp0s3      ethernet  connected     enp0s3
virbr0-nic  ethernet  disconnected  --
lo          loopback  unmanaged     --
```

Amint az első oszlopban láthatjuk, a hálózati eszközök listája. Van egy hálózati enp0s3 . A gépében más neveket is láthattál.

A név a hálózati kártya típusától függ (ha a fedélzeten van, pci kártya stb.). Az utolsó oszlopban a konfigurációs fájlokat láthatjuk, amelyeket eszközünk használ a hálózatra való csatlakozáshoz.

Egyszerűen érthető, hogy a mi készülékeink önmagukban nem tehetnek semmit. Szükségünk van arra, hogy készítsenek egy konfigurációs fájlt, hogy elmondhassuk nekik, hogyan érik el a hálózati kapcsolatot. Ezeket a fájlokat "kapcsolatprofiloknak" nevezzük. Megtaláljuk őket a / etc / sysconfig / network-scripts könyvtárban.

```
# cd /etc/sysconfig/network-scripts/
# ls
```

```
fcfg-enp0s3  ifdown-isdn      ifup           ifup-plip      ifup-tunnel
ifcfg-lo      ifdown-post      ifup-aliases   ifup-plusb      ifup-wireless
ifdown        ifdown-ppp       ifup-bnep      ifup-post       init.ipv6-
lobal
ifdown-bnep   ifdown-routes    ifup-eth       ifup-ppp        network-
functions
ifdown-eth    ifdown-sit       ifup-ib        ifup-routes     network-
```

```
functions-ipv6
ifdown-ib      ifdown-Team       ifup-ippp      ifup-sit
ifdown-ippp    ifdown-TeamPort   ifup-ipv6      ifup-Team
ifdown-ipv6    ifdown-tunnel     ifup-isdn      ifup-TeamPort
```

Amint itt látszik, az `ifcfg-` (interfész konfigurációval) kezdődő fájlok a kapcsolatprofilok. Amikor létrehozunk egy új kapcsolatot, vagy módosítunk egy meglévő nmcli vagy nmtui értékkel , akkor az eredményeket kapcsolatprofilként mentjük el. Megmutatom nekik kettőt a gépemről, egy dhcp konfigurációval és egy statikus IP-vel.

```
# cat ifcfg-static1
# cat ifcfg-Myoffice1
```

```
[root@ira network-scripts]# cat ifcfg-static1
TYPE=Ethernet
BOOTPROTO=none
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=static1
UUID=bfa7a232-f5b5-4d8f-a6fa-f972c60f56b7
DEVICE=enp0s3
ONBOOT=yes
IPADDR=192.168.1.40
PREFIX=24
GATEWAY=192.168.1.1
DNS1=8.8.8.8
DNS2=8.8.4.4
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

```
[root@ira network-scripts]# cat ifcfg-Myoffice1
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
NAME=Myoffice1
UUID=2a4a63ec-c6bd-4c76-a92e-0554b7c03817
DEVICE=enp0s3
ONBOOT=yes
PEERDNS=yes
PEERROUTES=yes
DHCP_HOSTNAME=ira
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
```

Tudjuk, hogy egyes tulajdonságok eltérő értékeket mutatnak, és mások nem léteznek, ha erre nincs szükség. Lássuk gyorsan a legfontosabbakat.

- `TYPE` , ethernet típusú itt van. Lehetne wifi, csapat, kötvény és mások.
- `DEVICE` , az ehhez a profilhoz társított hálózati eszköz neve.
- `BOOTPROTO` , ha értéke "dhcp", akkor kapcsolatprofilunk dinamikus IP-t kap a dhcp kiszolgálótól, ha értéke "none", akkor nem vesz dinamikus IP-t és valószínűleg statikus IP-t rendel.
- `IPADDR` , a statikus IP, amelyet hozzárendelünk profilunkhoz.
- `PREFIX` , az alhálózati maszk. A 24 érték 255.255.255.0 értéket jelent. Jobban megértheti az alhálózati maszkot, ha bináris formátumát írja le. Például a 16, 24, 26 értékek azt jelentik, hogy az első 16, 24 vagy 26 bitet 1, a többi 0 pedig pontosan meghatározza, hogy mi a hálózati cím, és mi az a tartomány, amelyik hozzárendelhető.

- GATEWAY , az átjáró IP.
- DNS1 , DNS1 , két dns szerver, amit használni akarunk.
- ONBOOT , ha értéke "igen", azt jelenti, hogy a rendszerindításkor a számítógép elolvassa ezt a profilt, és megpróbálja hozzárendelni a készülékéhez.

Most menjünk tovább, és ellenőrizzük kapcsolatainkat:

```
# nmcli con show
```

```
[root@ira network-scripts]# nmcli con show
NAME       UUID                                  TYPE           DEVICE
static1    bfa7a232-f5b5-4d8f-a6fa-f972c60f56b7  802-3-ethernet --
Myoffice1  2a4a63ec-c6bd-4c76-a92e-0554b7c03817  802-3-ethernet enp0s3
enp0s3     354faa8a-9efe-4cf0-a03b-d9dd0d7e6311  802-3-ethernet --
enp0s8     4c00d95c-7fa2-4e5b-968f-4b8609773501  802-3-ethernet enp0s8
```

Aktív hálózati kapcsolatok megjelenítése

Az eszközök utolsó oszlopa segít megérteni, hogy melyik kapcsolat "UP" és fut, és ami nem. A fenti képen láthatók a két aktív kapcsolat: Myoffice1 és enp0s8 .

Tipp : Ha csak az aktív kapcsolatokat szeretné látni, írja be:

```
# nmcli con show -a
```

Tipp : Az nmcli használatakor az automatikus kitöltés lapot használhatja, de jobb, ha a parancs minimális formátumát használja. Így a következő parancsok egyenlők:

```
# nmcli connection show
# nmcli con show
# nmcli cs
```

Ha ellenőrizem az eszközeim IP-címét:

```
# ip a
```

```
[root@ira network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:50:c4:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.6/24 brd 192.168.1.255 scope global dynamic enp0s3
       valid_lft 1811242sec preferred_lft 1811242sec
    inet6 2a02:582:5857:8400:a00:27ff:fe50:c419/64 scope global dynamic
       valid_lft 43890sec preferred_lft 43890sec
    inet6 fe80::a00:27ff:fe50:c419/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:33:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 1811239sec preferred_lft 1811239sec
    inet6 2a02:582:5857:8400:a00:27ff:fe33:2aaa/64 scope global dynamic
       valid_lft 43890sec preferred_lft 43890sec
    inet6 fe80::a00:27ff:fe33:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```

Ellenőrizze a szerver IP-címét

Látom, hogy az enp0s3 készülékem a 192.168.1.6 IP- enp0s3 a dhcp kiszolgálóról veszi át, mert a Myoffice1 kapcsolódási profil, Myoffice1 fel van állítva, dhcp konfigurációval rendelkezik. Ha a static1 nevű kapcsolati profilomat "up" static1 akkor a készülék az IP 192.168.1.40 statikus IP-címét veszi fel, amint azt a kapcsolatprofilban meghatározza.

```
  # nmcli con down Myoffice1;  nmcli con up static1
 # nmcli con show
```

Lássuk az IP címet újra:

```
  # ip a
```

```
[root@ira network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:50:c4:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.40/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 2a02:582:5857:8400:a00:27ff:fe50:c419/64 scope global dynamic
       valid_lft 86149sec preferred_lft 86149sec
    inet6 fe80::a00:27ff:fe50:c419/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:33:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 1810298sec preferred_lft 1810298sec
    inet6 2a02:582:5857:8400:a00:27ff:fe33:2aaa/64 scope global dynamic
       valid_lft 86149sec preferred_lft 86149sec
    inet6 fe80::a00:27ff:fe33:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```

Ellenőrizze a hálózati statikus IP-címet

Meg tudjuk csinálni az első kapcsolatprofilunkat. A minimális tulajdonságok meg kell adni a típus , ifnameés con-name :

- type - a kapcsolat típusához.
- ifname - a kapcsolat neve hozzárendelt eszköz neve.
- con-name - a kapcsolat nevéhez.

Csináljunk egy új ethernet kapcsolatot a Myhome1 nevű Myhome1 , amely az enp0s3 eszközhöz enp0s3 :

```
# nmcli con add type ethernet con-name Myhome1 ifname enp0s3
```

Ellenőrizze a konfigurációt:

```
  # cat ifcfg-Myhome1
```

```
[root@ira network-scripts]# nmcli con add type ethernet con-name Myhome1 ifname enp0s3
Connection 'Myhome1' (6ca1c9f5-0fb5-4505-a185-5e27b7fdeb14) successfully added.
[root@ira network-scripts]# cat ifcfg-Myhome1
TYPE=Ethernet
BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=Myhome1
UUID=6ca1c9f5-0fb5-4505-a185-5e27b7fdeb14
DEVICE=enp0s3
ONBOOT=yes
```

Új hálózati kapcsolat létrehozása

Mint látható, BOOTPROTO=dhcp , mert nem adtunk statikus IP címet.

Tipp : Az "nmcli con mod" paranccsal bármilyen kapcsolatot módosíthatunk. Ha azonban módosítja a dhcp-kapcsolatot, és statikusvá változtatja, ne felejtse el megváltoztatni az "ipv4.method" "auto" -ról "manual" . Ellenkező esetben két IP-címet fog kapni: az egyik a dhcp szerverről és a statikus.

Készítsünk egy új Ethernet kapcsolati profilt a static2 , amelyet az enp0s3 eszközhöz enp0s3 , statikus IP 192.168.1.50 , alhálózati maszk 255.255.255.0 = 24 és 192.168.1.1 átjáró.

```
# nmcli con add type ethernet con-name static2 ifname enp0s3 ip4
192.168.1.50/24 gw4 192.168.1.1
```

Ellenőrizze a konfigurációt:

```
# cat ifcfg-static2
```

```
[root@ira network-scripts]# nmcli con add type ethernet con-name static2 ifname enp0s3 ip4 192.168.1.50/24 gw4
 192.168.1.1
Connection 'static2' (40855e4f-c246-4947-8dc2-e37108c9ca97) successfully added.
[root@ira network-scripts]# cat ifcfg-static2
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.1.50
PREFIX=24
GATEWAY=192.168.1.1
DEFROUTE=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=static2
UUID=40855e4f-c246-4947-8dc2-e37108c9ca97
DEVICE=enp0s3
ONBOOT=yes
```

Új Ethernet kapcsolat létrehozása

Módosítsuk az utolsó csatlakozási profilt, és adjunk hozzá két dns szervert.

```
# nmcli con mod static2 ipv4.dns "8.8.8.8 8.8.4.4"
```

Tipp : Van itt valami, amit figyelni kell: az IP-cím és az átjáró tulajdonságai eltérő neveket adnak hozzá, és amikor módosítja a kapcsolatot. Amikor csatlakozásokat ad hozzá, az “ip4” és a “gw4” , miközben módosítod őket, az “ipv4” és a “gwv4” .

Most kapcsoljuk be ezt a kapcsolati profilt:

```
# nmcli con down static1;  nmcli con up static2
```

Amint láthatja, az eszköz enp0s3 most IP-címe 192.168.1.50 .

```
# ip a
```

```
[root@ira network-scripts]# nmcli con down static1 ; nmcli con up static2
Connection 'static1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnec
tion/8)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/10)
[root@ira network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:50:c4:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 2a02:582:5857:8400:a00:27ff:fe50:c419/64 scope global dynamic
       valid_lft 81106sec preferred_lft 81106sec
    inet6 fe80::a00:27ff:fe50:c419/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:33:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 1805254sec preferred_lft 1805254sec
    inet6 2a02:582:5857:8400:a00:27ff:fe33:2aaa/64 scope global dynamic
       valid_lft 81106sec preferred_lft 81106sec
    inet6 fe80::a00:27ff:fe33:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```

Ellenőrizze az új hálózati kapcsolat IP címét

Tipp : Rengeteg tulajdonságot lehet módosítani. Ha nem emlékszel rájuk szívesen, akkor segíthetsz magaddal az "nmcli con show" beírásával, majd ezt követően a kapcsolat nevét:

```
# nmcli con show static2
```


```
[root@ira network-scripts]# nmcli con down static1 ; nmcli con up static2
Connection 'static1' successfully deactivated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnec
tion/8)
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/10)
[root@ira network-scripts]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:50:c4:19 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.50/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
    inet6 2a02:582:5857:8400:a00:27ff:fe50:c419/64 scope global dynamic
       valid_lft 81106sec preferred_lft 81106sec
    inet6 fe80::a00:27ff:fe50:c419/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 08:00:27:33:2a:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.7/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 1805254sec preferred_lft 1805254sec
    inet6 2a02:582:5857:8400:a00:27ff:fe33:2aaa/64 scope global dynamic
       valid_lft 81106sec preferred_lft 81106sec
    inet6 fe80::a00:27ff:fe33:2aaa/64 scope link
       valid_lft forever preferred_lft forever
```


Ellenőrizze az új hálózati kapcsolat IP címét

A tulajdonságokat módosíthatja kisbetűkkel.

Például: ha összekapcsolási profilt hoz létre, a NetworkManager keres egy másik kapcsolódási profilt, és automatikusan megjeleníti azt. (Gyakorlatilag hagyom, hogy ellenőrizzem). Ha nem szeretné, hogy a kapcsolatprofil kapcsolódjon:

```
# nmcli con mod static2 connection.autoconnect no
```

Az utolsó gyakorlat nagyon hasznos: létrehozott egy kapcsolódási profilt, de azt szeretné, hogy bizonyos felhasználók használhassák. Jó osztályozni a felhasználókat!

A profil használatához csak a felhasználói stellát engedélyezzük:

```
# nmcli con mod static2 connection.permissions stella
```

Tipp : Ha egynél több felhasználónál engedélyt szeretne adni, be kell írnia a `user:user1,user2` között üres helyet:

```
# nmcli con mod static2 connection.permissions user:stella,john
```

```
[root@ira network-scripts]# nmcli con mod static2 connection.permissions user:stella,john
[root@ira network-scripts]# nmcli con show static2
connection.id:                          static2
connection.uuid:                        40855e4f-c246-4947-8dc2-e37108c9ca97
connection.interface-name:              enp0s3
connection.type:                        802-3-ethernet
connection.autoconnect:                 no
connection.autoconnect-priority:        0
connection.timestamp:                   1456649639
connection.read-only:                   no
connection.permissions:                 user:stella,user:john
connection.zone:                        --
connection.master:                      --
connection.slave-type:                  --
connection.secondaries:
connection.gateway-ping-timeout:        0
802-3-ethernet.port:                    --
802-3-ethernet.speed:                   0
802-3-ethernet.duplex:                  --
802-3-ethernet.auto-negotiate:          yes
802-3-ethernet.mac-address:             --
802-3-ethernet.cloned-mac-address:      --
802-3-ethernet.mac-address-blacklist:
802-3-ethernet.mtu:                     auto
802-3-ethernet.s390-subchannels:
802-3-ethernet.s390-nettype:            --
802-3-ethernet.s390-options:
ipv4.method:                            manual
ipv4.dns:                               8.8.8.8,8.8.4.4
ipv4.dns-search:
ipv4.addresses:                         192.168.1.50/24
ipv4.gateway:                           192.168.1.1
ipv4.routes:
```

Hálózati kapcsolatok engedélyezése a felhasználók számára

Ha bejelentkezik egy másik felhasználónak, akkor nem hozhatja "fel" ezt a kapcsolati profilt:

```
 # nmcli con show
# nmcli con up static2
# ls / etc / sysconfig / network-scripts
```

```
[artemis@ira Desktop]$ nmcli con show
NAME       UUID                                  TYPE            DEVICE
Myoffice1  2a4a63ec-c6bd-4c76-a92e-0554b7c03817  802-3-ethernet  --
Myhome1    6ca1c9f5-0fb5-4505-a185-5e27b7fdeb14  802-3-ethernet  --
enp0s8     4c00d95c-7fa2-4e5b-968f-4b8609773501  802-3-ethernet  enp0s8
static1    bfa7a232-f5b5-4d8f-a6fa-f972c60f56b7  802-3-ethernet  enp0s3
[artemis@ira Desktop]$ nmcli con up static2
Error: Connection 'static2' does not exist.
[artemis@ira Desktop]$ ls /etc/sysconfig/network-scripts/
ifcfg-enp0s3     ifcfg-static2  ifdown-ipv6    ifdown-Team      ifup-eth   ifup-plusb   ifup-TeamPort
ifcfg-enp0s8     ifdown         ifdown-isdn    ifdown-TeamPort  ifup-ib    ifup-post    ifup-tunnel
ifcfg-lo         ifdown-bnep    ifdown-post    ifdown-tunnel    ifup-ippp  ifup-ppp     ifup-wireless
ifcfg-Myhome1    ifdown-eth     ifdown-ppp     ifup             ifup-ipv6  ifup-routes  init.ipv6-global
ifcfg-Myoffice1  ifdown-ib      ifdown-routes  ifup-aliases     ifup-isdn  ifup-sit     network-functions
ifcfg-static1    ifdown-ippp    ifdown-sit     ifup-bnep        ifup-plip  ifup-Team    network-functions-ipv6
[artemis@ira Desktop]$
```

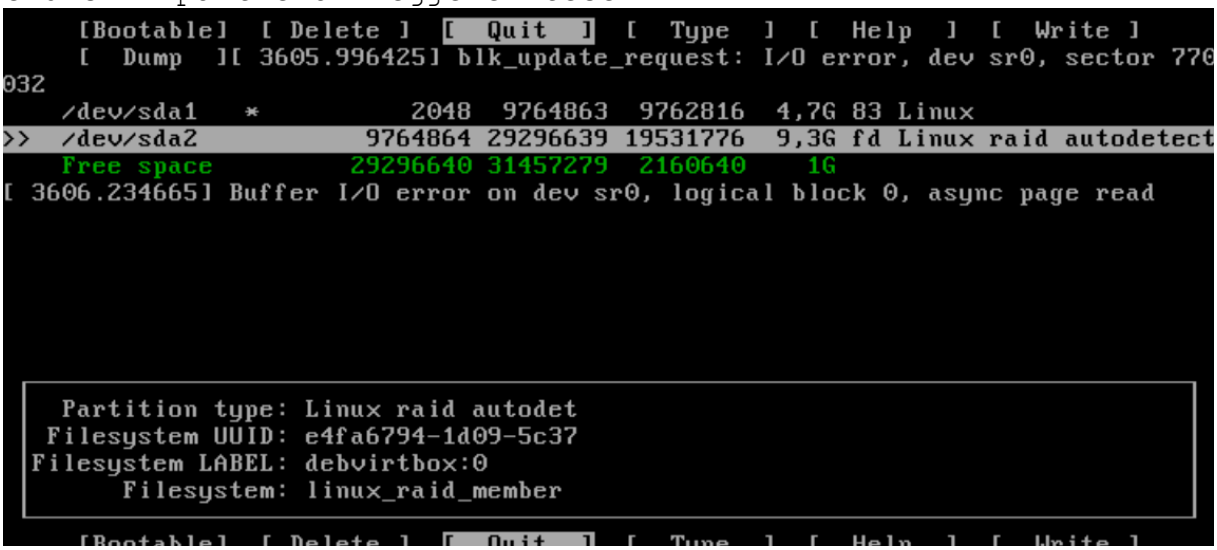
Engedélyezze a hálózati kapcsolatot

Egy hibaüzenet azt mondja, hogy a "static2" kapcsolat nem létezik , még akkor sem, ha látjuk, hogy létezik.Ez azért van, mert a jelenlegi felhasználónak nincs engedélye a kapcsolat létrehozására.

Következtetés : ne habozzon az nmcli használatával. Könnyű és hasznos.

Partíciók ellenőrzése

cfdisk – particiok megjelenitese



Alul egy menü találunk, amiben a „jobbra” és „balra” billentyűkkel mozoghatunk. A kilépés is itt választható. Minden partíciónak van egy alaptípusa, ezt lehet a [Type] menüben beállítani.

Ha elkészültek a partíciók, a végén ki kell a változásokat írni a háttértárra. Ezt a [Write] paranccsal tehetjük meg. Ügyeljünk arra, hogy rákérdez, biztosan szeretnénk-e végrehajtani a kiírást. Itt a „yes” szót kell begépelni, nem elég a „y” önmagában.

Ha két merevlemez van egy számítógépben és azok SATA csatlakozóval rendelkeznek, akkor az első merevlemez neve sda, a második merevlemez neve sdb

- /dev/sda
- /dev/sdb

Ha elindítjuk a cfdisk particionáló programot paraméter nélkül, akkor az első merevlemezt tudjuk szerkeszteni. Ha másodikat szeretnénk, akkor meg kell adni paraméterként, például így:

cfdisk /dev/sdb

Ha kiléptünk a cfdisk programból, érdemes az eredményt a fidsk -l paranccsal is megtekinteni.

Amennyiben még sosem volt particionálva a egy merevlemez a cfdisk felkínálja, hogy válasszunk partíciós címként. A következő lehetőségek vannak.

- gpt
- dos
- sgi
- sun

Általában a dos vagy gpt használatos.

Parancsok

df -h Mennyi szabad helyunk van a rendszeren
df -T Lássam a fájl rendszert is

SAMBA fájlmegosztás

Telepítése: **`apt-get install samba`**

Fő konfigfájl: **`/etc/samba/smb.conf`**

Specialitásai:

- nem csak a # hanem a ; is kommentet vezet be.
- minden csatlakozás külön kiszolgálói folyamatot indít (smbd)
- mindig van egy szabad kiszolgálói folyamat (smbd)
- mindig fut egy névkiszolgáló folyamat (nmbd)
  - a beállítások érvénybe lépéséhez nem kell újraindítani; a következő csatlakozásnál az új beállítások működnek
  - a beállítások a folyamatban lévő kapcsolatokat nem befolyásolják

Fő konfigfájl felépítése:

[szekciónév]
szekcióparaméter = érték
Fő szekciói:
[global] ? a teljes smb működésére kihat
[fenykepek] ? csak a „fenykepek” megosztásra érvényes

Samba felhasználó

A samba szerveren külön létre kell hozni az smb-felhasználókat.
Ezt az **`smbpasswd -a USER`** –rel tudjuk megtenni (a USER-nek már linuxosan léteznie kell). Ilyenkor bekéri a USER sambás jelszavát.

```
smbpasswd -a krisztian
```

A létező samba userek letilthatók (**`smbpasswd -d USER`**), engedélyezhetők (**`smbpasswd -e USER`**).

A USER sambás jelszavának megváltoztatása **`smbpasswd USER`** –rel történhet.

A kiszolgáló kipróbálása helyben a legegyszerűbb, ehhez SMB kliensprogram kell.

Home mappa megosztása

nano /etc/samba/smb.conf

```
#======================= Share Definitions =======================

[homes]
   comment = Home Directories
   browseable = yes

# By default, the home directories are exported read-only. Change the
# next parameter to 'no' if you want to be able to write to them.
   read only = no
```

Más mappa megosztása

A config fájl végére kell létrehozni
mkdir /var/samba létrehozom a megosztandó mappát

```
[samba]
path = /var/samba
writeable = yes
browseable = yes
valid users = krisztian
```

/etc/init.d/samba restart
/etc/init.d/smbd restart nem szükséges újraindítani(de nem is baj)

A windows 10 csodája…

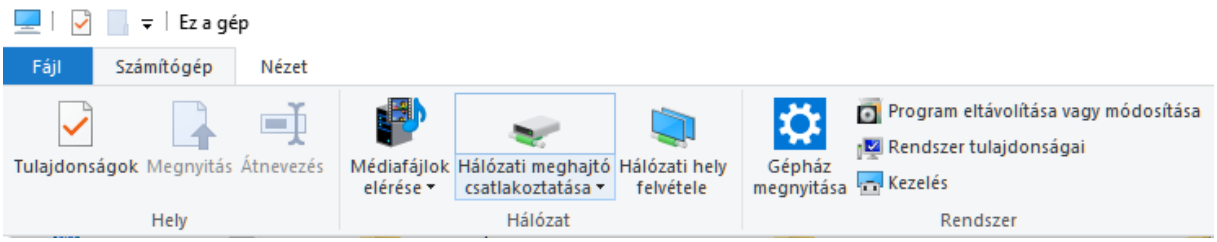
Nyisd meg a Csoportházirend szerkesztőt (rákeresel, vagy futtatásba írd be: **gpedit.msc**)

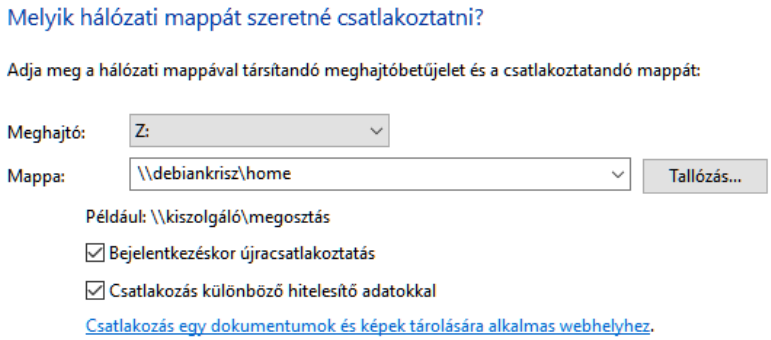
Navigálj el ide: Számítógép konfigurációja > Felügyeleti sablonok > Hálózat > Lanman-munkaállomás

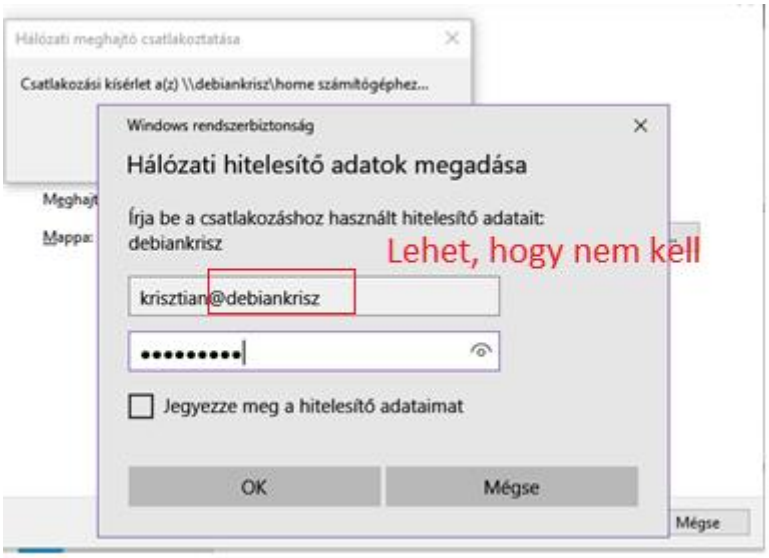
Dupla kattintás a Vendégek nem biztonságos bejelentkezéseinek engedélyezése bejegyzésen és válaszd az Engedélyezve opciót, majd OK

Majd futtasd a **gpupdate** parancsot (startmenü vagy cmd), hogy biztosan érvényben legyenek a fenti beállítások.

Megosztott mappa felcsatolása meghajtóként Windows 10-re







Megosztott mappa cstlakozása linuxra

apt-get install samba-client

használata:

smbclient UNC paraméterek

vagy

smbclient paraméterek UNC

Hálózati mappa böngészése: **smbclient \\\\ipcím\\megosztás -U felhasználó**

```
root@debiankrisz:/home/krisztian# smbclient \\\\192.168.0.143\\megosztva -U Krisztian
Enter Krisztian's password:
Domain=[DESKTOP-T6A2HJI] OS=[Windows 10 Enterprise 16299] Server=[Windows 10 Enterprise 6.
]
smb: \>
```

VAGY

smbclient //ipcím/megosztás -U felhasználó

SMB megosztás csatolása a fájlrendszerbe

Ehhez a CIFS összetevő szükséges: **apt-get install cifs-utils**

Hálózati mappa csatolása (a csatoláshoz helyi jogok kellenek ? root):

root@debiankrisz:/home/krisztian# mkdir /home/krisztian/Dokumentumok/windowsgep //létrehozom a cstolás helyét

root@debiankrisz:/home/krisztian# chmod 777 /home/krisztian/Dokumentumok/windowsgep //jogosultságok állítása

root@debiankrisz:/home/krisztian# mount.cifs //192.168.0.143/megosztva -o username=Krisztian /home/krisztian/Dokumentumok/windowsgep //felcsatolom a mappát a windows gépen lévő felhasználóval

Password for Krisztian@//192.168.0.143/megosztva: ******** //Megadom a windows gépen lévő felhasználó jelszavát

```
root@debiankrisz:/home/krisztian# mkdir /home/krisztian/Dokumentumok/windowsgep
root@debiankrisz:/home/krisztian# chmod 777 /home/krisztian/Dokumentumok/windowsgep
root@debiankrisz:/home/krisztian# mount.cifs //192.168.0.143/megosztva -o username=Krisztian /home/krisztian/Dokumentumok/windowsgep
Password for Krisztian@//192.168.0.143/megosztva:  ********
root@debiankrisz:/home/krisztian#
```

Jogosultsági kérdések:

- tartományi felhasználónevet **TARTOMÁNY\\USERNÉV** formában kell megadni
- a mappára a megadott felhasználónak jogokkal kell rendelkeznie (akár a listázáshoz is!!!)
- a megosztásra a megadott felhasználónak jogokkal kell rendelkeznie (akár a listázáshoz is!!!)

Syslog szerver készítése

Az **rsyslog** keresése a tárolóban

```
root@debiankrisz:/home/krisztian# apt-cache search syslog | grep rsys
fusiondirectory-plugin-rsyslog - rsyslog plugin for FusionDirectory
gosa-plugin-rsyslog - rsyslog plugin for GOsa²
rsyslog - reliable system and kernel logging daemon
rsyslog-elasticsearch - Elasticsearch output plugin for rsyslog
rsyslog-gnutls - TLS protocol support for rsyslog
rsyslog-gssapi - GSSAPI authentication and encryption support for rsyslog
rsyslog-mongodb - MongoDB output plugin for rsyslog
rsyslog-mysql - MySQL output plugin for rsyslog
rsyslog-pgsql - PostgreSQL output plugin for rsyslog
rsyslog-relp - RELP protocol support for rsyslog
rsyslog-doc - documentation for rsyslog
root@debiankrisz:/home/krisztian#
```

`apt-get install rsyslog` //feltelpítem a programot

`apt-get install nmap` //hálózat figyelő

`namp localhost` //hogy mely portok vannak nyitva a TCP portok küzül

```
root@debiankrisz:/home/krisztian# nmap localhost

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-15 16:54 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000012s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 993 closed ports
PORT    STATE SERVICE
21/tcp  open  ftp
22/tcp  open  ssh
25/tcp  open  smtp
80/tcp  open  http
111/tcp open  rpcbind
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Nmap done: 1 IP address (1 host up) scanned in 2.45 seconds
root@debiankrisz:/home/krisztian#
```

namp localhost -sU //hogy mely portok vannak nyitva az UDP portok küzül

```
root@debiankrisz:/home/krisztian# nmap localhost -sU

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-15 16:57 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000014s latency).
Other addresses for localhost (not scanned): 127.0.0.1
Not shown: 995 closed ports
PORT     STATE         SERVICE
68/udp   open|filtered dhcpc
111/udp  open          rpcbind
137/udp  open          netbios-ns
138/udp  open|filtered netbios-dgm
5353/udp open|filtered zeroconf
```

namp localhost –sU –p 510-520 //ha konkrét portot keresek

```
root@debiankrisz:/home/krisztian# nmap localhost -sU -p 510-520

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-15 17:00 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000043s latency).
Other addresses for localhost (not scanned): 127.0.0.1
PORT    STATE  SERVICE
510/udp closed fcp
511/udp closed passgo
512/udp closed biff
513/udp closed who
514/udp closed syslog
515/udp closed printer
516/udp closed videotex
517/udp closed talk
518/udp closed ntalk
519/udp closed utime
520/udp closed route
```

Mivel az **514**-es udp port nincs nyitva (ez kellene a syslog-hoz), ezért ki kell nyitni.

nano /etc/rsyslog.conf //kiveszem a kommentet az alábbi három sor elöl

```
provides UDP syslog reception
$ModLoad imudp
$UDPServerRun 514
```

/etc/init.d/rsyslog restart
namp localhost –sU –p 510-520 //és már nyitva is

```
root@debiankrisz:/home/krisztian# /etc/init.d/rsyslog restart
[ ok ] Restarting rsyslog (via systemctl): rsyslog.service.
root@debiankrisz:/home/krisztian# nmap localhost -sU -p 510-520

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-15 17:07 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000038s latency).
Other addresses for localhost (not scanned): 127.0.0.1
PORT    STATE         SERVICE
510/udp closed        fcp
511/udp closed        passgo
512/udp closed        biff
513/udp closed        who
514/udp open|filtered syslog
515/udp closed        printer
516/udp closed        videotex
517/udp closed        talk
518/udp closed        ntalk
```

A **/var/log/syslog** fáljba fog kerülni minden bejegyzés.

A cisco ruteren:

loggingon
logging host 192.168.0.227 /a linux gép ip címe
logging trup debugging

Az üzenet keresése: **cat /var/log/syslog | grep 192.168.0.129**

```
root@debiankrisz:/home/krisztian# cat /var/log/syslog | grep 192.168.0.129
Apr 15 17:13:28 192.168.0.129 41: *Apr 15 16:03:03.579: %SYS-6-LOGGINGHOST_START
STOP: Logging to host 192.168.0.227 port 514 started - CLI initiated
root@debiankrisz:/home/krisztian#
```

És itt is van a bejegyzéss.

Linux parancsok

Saját parancs készítése

alias dff=”df -h” Sajat parancs letrehozasa. Innentol mar a dff fogja jelenteni a df -h parancsot „ HELYETT APOSZTROF

unalias dff a definialt parancsom torlese

Ha a .bashlc ben átírok egy-egy parancsot, majd leveszem az írás jogot a többi felhasznalótól, akkor el tudok fedni parancsokat is. Így Ők ezeket a paracsokat már nem tudják használni.

Sources list szerkesztése

 # nano /etc/apt/source-list

```
# deb cdrom:[Debian GNU/Linux 9.3.0 _Stretch_ - Official i386 NETINST 20171209-$

#deb cdrom:[Debian GNU/Linux 9.3.0 _Stretch_ - Official i386 NETINST 20171209-1$

deb http://ftp.bme.hu/debian/ stretch main
deb-src http://ftp.bme.hu/debian/ stretch main

deb http://security.debian.org/debian-security stretch/updates main
deb-src http://security.debian.org/debian-security stretch/updates main

# stretch-updates, previously known as 'volatile'
deb http://ftp.bme.hu/debian/ stretch-updates main
deb-src http://ftp.bme.hu/debian/ stretch-updates main
```

Patakyban:
deb http://192.168.19.222/debian stretch main contrib non-free

Olvashatósági teszt

dd if=/dev/sda of=/dev/null a komplett merevlemezt blokkonként bemásolja a „semmibe” → olvasási teszt!

dd if=/dev/sda of=/dev/sdb → klónozom a merevlemezt az sdb-re
dd if=/dev/sda of=/backup/mentes.iso keszitek egy iso fajlt a komplett merevlemezemrol

Felhasználók kezelése

adduser – felhasználó létrehozása

Az adduser csak Debian alapú rendszereken létezik (az rpm alapú rendszereknél egy useradd egy alisa szokott lenni). Egy felhasználót interaktívan vehetünk fel vele. A következő példában egy kati nevű felhasználót veszünk fel:

```
adduser kati
Adding user `kati' ...
Adding new group `kati' (1002) ...
Adding new user `kati' (1002) with group `kati' ...
Creating home directory `/home/kati' ...
Copying files from `/etc/skel' ...
Adja meg az új UNIX jelszót:
Írja be újra a UNIX jelszót:
passwd: a jelszó sikeresen frissült
kati felhasználói információinak cseréje
Add meg az új értéket vagy üss ENTER-t az alapértelmezetthez
        TELJES Név []: Teszt Katalin
        Szobaszám []:
        Munkahelyi telefon []:
        Otthoni telefon []:
        Egyéb []:
Is the information correct? [Y/n] y
```

A program megáll először a jelszóbekérésnél, majd az egyéb adatok bekérésénél, a végén rákérdez, hogy az adatok rendben vannak-e. Az y vagy egy Enter lenyomása után a felhasználó létrejött könyvtárával együtt. Sőt a /etc/skel könyvtár tartalmát is megkapta.

Így is használhatjuk:

```
adduser --home /home/kati --shell /bin/bash --gecos "Teszt Katalin"
```

getent - információ

Információt szolgáltat egy adatbázisból. Olyan adatbázisból mint passwd, group

```
getent passwd joska
joska:x:1001:1001:Nagy József,,,:/home/joska:/bin/bash
```

A joska csoportról szeretnénk informálódni:

```
getent group joska
```

useradd

A useradd parancs minden linuxos rendszer része. Ezzel is felhasználókat tudunk felvenni, de mindent kapcsolókkal kell megadnunk.

```
useradd -c "Teszt Katalin" -d /home/kati -g users -G info,human,rgazda -k
/etc/skel -m -s /bin/bash kati
```

A useradd kapcsolói	
-c	megjegyzés
-d	home könyvtár
-g	elsődleges csoport
-G	másodlagos csoport vagy csoportok
-m	hozzuk létre a home könyvtárát
-s	milyen shellt kapjon
-u UID	mi legyen a uid-je
-k	honnan másoljuk alapértelmezett fájlokat

chage - Felhasználói jelszó lejárása

A joska felhasználónak mikor jár le a jelszava?

```
chage -l joska
chage -l joska
Utolsó jelszóváltás                                     : dec 13, 2011
Jelszó lejár                                        : soha
Jelszó inaktív                                      : soha
Hozzáférés lejár                                                : soha
A jelszómódosítások közti legkevesebb nap               : 0
A jelszómódosítások közti legtöbb nap           : 99999
A jelszó lejárata előtt figyelmeztetés napok száma      : 7
```

99999 nap körülbelül: ~273 év

A joska felhasználó jelszava járjon le 10 nap múlva

```
chage -M 10 joska
```

Mikor járjon le a jelszava?

```
chage -E "2009-05-31" joska
```

Inaktív napok után lezárás

Jóskának 10 inaktív nap után lezárjuk a jelszavát.

```
chage -I 10 joska
chage --inactive 10 joska
```

Alapértelmezés visszaállítása

Az alábbi kapcsolókat használjuk:

```
-m 0 A napok minimális száma amíg lejár a jelszó.
-M 99999 A jelszó maximum 99999 nap múlva jár le.
-I -1 (minusz egy) A jelszó sosem lesz inaktív.
-E -1 (minusz egy) A fiók sosem jár le.
chage -m 0 -M 99999 -I -1 -E -1 joska
```

Figyelmeztetés

Figyelmeztetés mielőtt a jelszó lejár

- -W, --warndays WARN_DAYS

```
chage -W 7 joska
```

addgroup – c soport létrehozása

Az addgroup valójában egy link az adduser parancsra. Kézikönyve is megegyezik vele.

Csoport felvétele:

```
addgroup info
```

groupadd - Csoport felvétele

Ha egy felhasználót felvettünk egy csoportba, a csoport tagsága csak az újbóli belépés után lesz érvényes.

A groupadd parancs minden linuxos rendszerben megtalálható.

Van egy másik csoportok felvételére kitalált parancs, ez az addgroup. A groupadd parancstól csak annyiban különbözik, hogy tájékoztatást ír a képernyőre két sorban a felvétel sikeréről.

Például a „human” nevű csoport felvétele:

```
groupadd human
```

vagy:

```
addgroup human
Adding group `human' (GID 1003) ...
Kész.
```

gpasswd

A gpasswd parancs a /etc/group állomány adminisztrálására lett megalkotva.

A „kati” nevű felhasználó felvétele a „human” csoportba:

```
gpasswd -a kati human
```

A „kati” nevű felhasználó törlése a „human” csoportból:

```
gpasswd -d kati human
```

usermod – felhasználó csoporthoz adása

A mari felhasználó felvétele a human csoportba:

```
usermod -a -G human mari
```

A -a hatására hozzáfűzés történik, vagyis a többi csoport megmarad. Ha elhagyjuk a -a kapcsolót, akkor a többi csoport törlődik.

```
usermod -G human,gazdasag,rgazda mari
```

A joska felhasználó **kitiltása**:

```
usermod --lock --expiredate 1970-01-01 joska
usermod -L -e 1970-01-01 joska
```

Az expiredate értéke bármi lehet ami régebbi mint az aktuális dátum.

A --lock a jelszó használatot tiltja. Azonban nem tiltja a kulcs alapú azonosítást. Ezért vettük vissza lejárati dátumot.

Magunk is letilthatjuk a felhasználó jelszavas bejelentkezését ha egy „!” („felkiáltó jelet”) teszek a jelszó elé:

```
joska:!$1$R8Z4PoEr$OSi234nDisere2ERda83dD82DSIK8ls:15407:0:99999:7:::
```

A felhasználó számára új bejelentkezési könyvtárat állítunk be:

```
usermod -d /home/info/joska
usermod --home /home/info/joska
usermod --home /home/tanulok/15z/janos
```

Alap csoport beállítása:

```
usermod -g info mari
```

Shell beállítása:

```
usermod -s /bin/ksh mari
usermod --shell /bin/ksh kati
```

A jelszót nem lehet lecserélni adott ideig:

- passwd -n MIN <login-name>

```
# passwd -n 10000 janos
```

id

Ha egy felhasználó belépett saját csoporttagságairól az id parancs segítségével tájékozódhat.

A parancs önmagában kiadva is hatásos:

```
id
```

De lássuk paraméterezve.

Az aktuális felhasználó milyen csoportokban van benne:

```
id -nG
```

Az adott felhasználó milyen csoportokban van benne:

```
id -nG mari
```

vipw

A passwd, illetve a shadow fájl szerkesztése. Tulajdonképpen a vi szövegszerkesztőben nyitja meg a /etc/passwd fájl tartalmát.

Az EDITOR környezeti változóval más szerkesztő is megadható. Például az mcedit használata:

```
export EDITOR=mcedit; vipw
```

vigr

A csoportfájl szerkesztése.

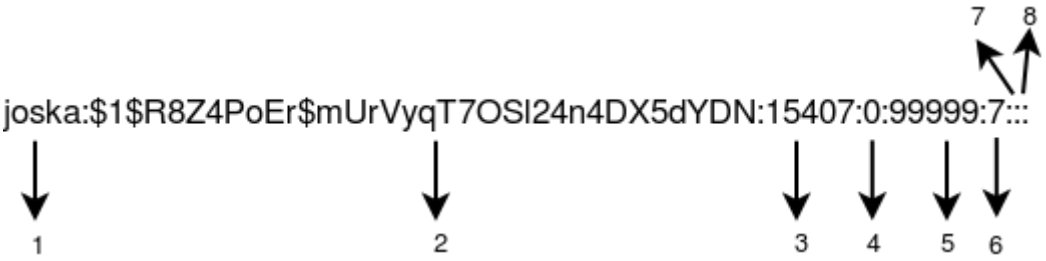
Mint a vipw, csak a csoportfájlt szerkesztjük.

A shadow fájl

A régi unixos rendszerekben a jelszavak a passwd fájlban voltak. Ebben a fájlban vannak tárolva a felhasználók egyéb adatai, mint teljes név, szoba, stb. Ezek nyilvános adatok, így mindenki számára olvashatók. A Linuxokon felmerült az igény a jelszavak külön fájlban való elhelyezésére. Így került az /etc/shadow állományba. Az átlagos felhasználó ezeket nem tudja olvasni.

A /etc/shadow fájl tartalmazza a kódolt jelszót. Például

```
joska:$6$wR8ZPo/p$.XKKmA9YCSrK5UuA0BV03mUrVyqT7OSl24n4DX5dYDNolZDhD.YfE0Fn.PVz
aDDGrxMwK6x4jZVTz9/0R8jQs1:15407:0:99999:7:::
```



1. Felhasználónév
2. Jelszó
3. Utolsó jelszómódosítás és 1970 január 01-óta eltelt napok száma
4. minimum ennyi napnak kell eltelni az újabb jelszómódosításig
5. maximum ennyi ideig érvényes a jelszó
6. Figyelmeztetés ennyi nappal a jelszó lejárat előtt
7. Ha lejár a jelszó, ennyi nap múlva nem léphet be
8. 1970 január 01 után ennyi nap letelte után lejár a jelszó

Jelszógenerálás

Néha szükségünk lehet arra, hogy új rendszerünkbe a felhasználóknak jelszót generáljunk.

Egyik erre alkalmas program a makepasswd. Használatához telepíteni kell a makepasswd csomagot:

```
apt-get install makepasswd
```

A makepasswd alapértelmezetten 8 és 10 karakter közötti jelszót generál. Kapcsolóval szabályozhatjuk hány karakteres legyen. Például nyolc karakteres:

```
makepasswd --chars 8
```

Ha több jelszót szeretnénk generálni, a count kapcsolóval megtehetjük.

```
makepasswd --chars 18 --count 10
```

A jelszó kódolt formájának generálása az Apró debianos tippek oldalon található.

Másik jelszógeneráló program:

```
apt-get install passwdqc
```

Jelszó generálása:

```
pwqgen
```

Felhasználó törlése

userdel

Töröljük a felhasználót:

```
userdel joska
```

Töröljük a felhasználót és a könyvtára tartalmát:

```
userdel -r joska
```

deluser

```
deluser --remove-home

deluser --remove-all-files

deluser --backup

deluser --backup-to
```

Az /etc/deluser.conf segítségével érdemes szabályozni. Részletekért nézzük meg a deluser.conf(5) kézikönyvet.

```
--backup
```

Minden állomány backupja, ami a home és a mailspool könyvtárban van. Az eredmény:

```
/$user.tar.bz2 vagy /$user.tar.gz.

--backup-to
```

A backup fájl helyét is megadhatjuk

```
--remove-home
```

A felhasznál mailspool és home könyvtárának törlése. Ha a --backup meg van adva, akkor backup után a fájlok törlésre kerülnek.

```
--remove-all-files
```

Minden fájl törlése a rendszerben, amelynek a felhasználó a birtokosa. Ha a --backup meg van adva, a törlés a backup után történik.

chfn

A finger információ (teljes név, iroda, telefon, stb) módosítása.

```
# chfn -f János janos
# chfn --full-name "Nagy János" janos
$ getent passwd janos
```

Kapcsolók:

- -f, --full-name
- -o, --office -- irodai szobaszám
- -p, --office-phone -- irodai telefonszám
- -h, --home-phone -- otthoni telfonszám

finger

```
apt install finger
finger joska
```

passwd

Használat például:

```
passwd  \
    -x <MAX_D> \
    -w <WARN_D> \
    -i <INACTIVE_D>
   felhasználó
```

A passwd parancs a jelszó beállítására használható.

```
# passwd
```

A rendszergazda mások jelszavát is beállíthatja:

```
# passwd janos
```

A felhasználó kizárása:

```
# passwd -l janos
```

A felhasználó újból engedése:

```
# passwd -u janos
```

Jelszó lejárat előtt hány nappal kapjon a felhasználó figyelmeztetést:

```
# passwd -w 7 janos
```

A felhasználó jelszavának lejárata előtt figyelmeztetés 7 nappal.

A felhasználó 7 napig nem változtathatja meg a jelszót:

```
# passwd -n 7
```

Jelszóbeállítások lekérdezése:

```
# passwd –S
# passwd -S janos
```

A felhasználó jelszavának azonnali „lejáratása”:

```
# passwd -e janos
```

Adott felhasználó jelszavának törlése:

```
# passwd -d janos
```

További kapcsolók:

```
# passwd -h
```

login beállítások

Az /etc/login.defs könyvtárban a felhasználók számára beállítható néhány adottság. Ilyen a felhasználók levelei hol tárolódjanak.

```
MAIL_DIR /var/mail
```

A sikertelen bejelentkezések naplózása a /var/log/faillog fájlba.

```
FAILLOG_ENAB yes
```

Az ismeretlen felhasználónevek naplózása sikertelen bejelentkezés esetén.

```
LOG_UNKFAIL_ENAB no
```

A sikeres bejelentkezések naplózása:

```
LOG_OK_LOGINS no
```

A su tevékenységek naplózása a syslog naplóba:

```
SYSLOG_SU_ENAB yes
SYSLOG_SG_ENAB yes
```

A su aktívitás külön naplózása:

```
SULOG_FILE /var/log/sulog
```

Alapértelmezett útvonal a rendszergazda számára:

```
ENV_SUPATH
```

Alapértelmezett útvonal a felhasználók számára:

```
ENV_PATH
```

Jelszavak kontrollálása. Alapértelmezésként, mikor jár le, mikor változtathatja meg, mikor legyen figyelmeztetés.

```
PASS_MAX_DAYS 99999
PASS_MIN_DAYS 0
PASS_WARN_AGE 7
```

Felhasználók minimális azonosítója useradd esetén:

```
UID_MIN 1000
UID_MAX 60000
```

Minimális csoportazonosítók:

```
GID_MIN 1000
GID_MAX 60000
```

A bejelentkezés visszautasítása ennyi sikertelen próbálkozás után:

```
LOGIN_RETRIES 5
```

A maximális sikertelen bejelentkezések után ennyi ideig tiltva:

```
LOGIN_TIMEOUT 60
```

További lehetőségek a /etc/login.defs állományban is találhatók, vagy nézzük meg az idevonatkozó kézikönyvet:

```
man login.defs
```

A felhasználók a chfn paranccsal milyen értékeket cserélhetnek:

```
CHFN_RESTRICT rwh
```

- f - full name
- r - room number

- w - work phone
- h - home phone

Ha a felhasználó nem tud belépni a könyvtárába, akkor legyen egy alapértelmezett könyvtár:

```
DEFAULT_HOME yes
```

A felhasználó törlésekor a következő parancs fusson le:

```
USERDEL_CMD /usr/local/bin/sajtScript.sh
```

Felhasználónévvel azonos csoport létrehozása useradd, törlése userdel esetén:

```
USERGROUPS_ENAB yes
```

Az aktuális shell előtt milyen parancs fusson le:

```
FAKE_SHELL /bin/fakeshell
```

Azonosítás algoritmusa:

```
ENCRYPT_METHOD SHA512
```

Felhasználócsere

A felhasználót a su paranccsal cserélhetünk. Például mari felhasználóvá válhatunk:

```
su mari
```

Ha kötőjelet is használok, akkor a mari felhasználó teljes környezetét kapom:

```
su - mari
```

Ehhez a művelethez persze tudni kell a mari felhasználó jelszavát, vagy rendszergazdaként kell végrehajtani.

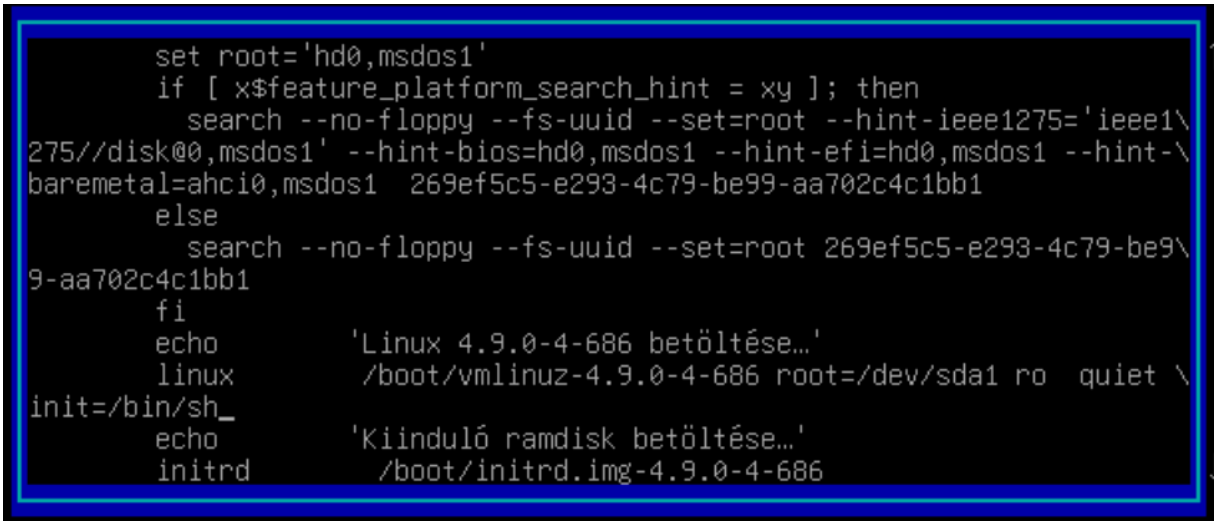
Egy csoportot is felvehetünk a sg paranccsal. Például az info csoport felvétele:

```
sg info
```

Linux Elfelejtett jelszó helyreállítása

Megállítóm a GRUB folyamatot, majd lenyomom az **e** bgombot.

A linux kezdetű sor végére beírom: **init=/bin/sh**

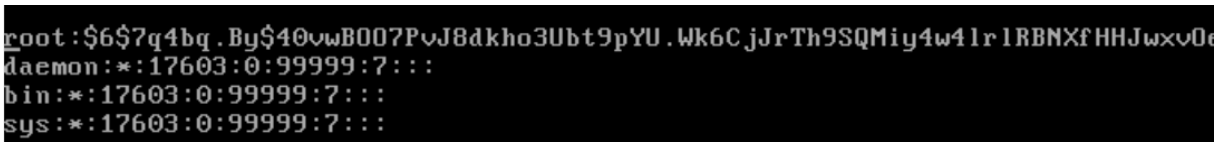


Majd**ctrl+x**-el továbbengedet a GRUB folyamatot, de így már csak egy alap bash fog elindulni.

Újra felcsatolom az egész féjrendszert ovashatóként:

mount –orw,remount /

nano /etc/shadow fáljban vannak a jelszavak titkosítva tárolva

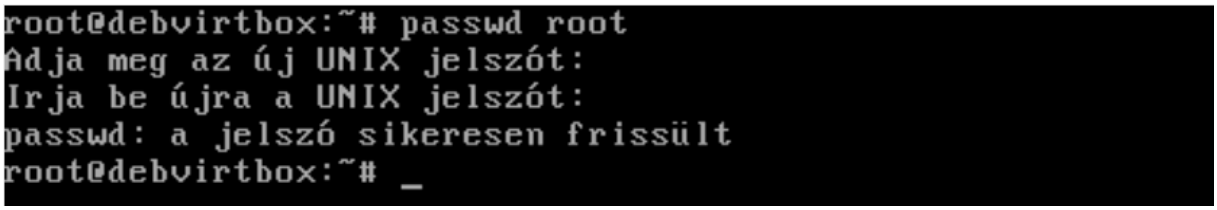


itt kitörlöm a root utáni két kettőspont közötti részét

sync, hogy az adatok a háttértáron biztosan szinkronizálva legyenek.

mount –oro,remount / így már csak olvasható lesz a fájlrendszer újra.

Majd újraindítom és belépek **root**ként és nem kér jelszót. Ezért adopk neki egy root jelszót:



Jogosultságok kezelése

A Linux operációs rendszereken elengedhetetlen a jogosultságok használata, hiszen így megakadályozhatjuk, hogy mások esetleg hozzáférjenek a dokumentumainkhoz vagy a többi adatunkhoz. Meg tudjuk szabni, hogy ki olvashatná, esetleg módosíthatná vagy akár futtathatná a fájlunkat. Most már tudjátok, hogy a Linux operációs rendszerek egyik alappillére a jogosultság. Most pedig dőljünk hátra és tudjuk meg mire is jó és hogyan alkalmazzuk a jogosultságokat a rendszerünkön.

Mint ígértem, igaz kicsit megkésve, de most volt időm megírni a jogosultságokról szóló cikket. :)

Akkor csapjunk is bele a lecsóba…

Tudjuk, hogy

- minden állománynak van tulajdonosa, csoportja és mindenki más
- mindezekhez tartozik olvasási (**r**), írási (**w**) és futtatási (**x**) jog
- a fájl(ok) futtatásához **rx** kell és a mappa megnyitáshoz is szintén **rx**kell

Ezek a betűk, pontosabban az **rwx** trió, mind mind egy-egy angol szónak a betűi, és mivel olvasásról, írásról és futtatásról beszélünk, így ezeknek a betűknek az angol megfelelője **R**ead, **W**rite, e**X**ecute.

Nézzünk egy táblázatot erről, betűkkel (**rwx**):

Jogosultság – **`chmod`** *parancs*			
Engedélyek	Tulajdonos	Csoport	Mindenki más
Olvasás	r	r	r
Írás	w	w	w
Futtatás	x	x	x
Összegezve:	rwx	rwx	rwx

Ha egy fájlnak a jogosultsága “**-rw-r–r–**” akkor a következőképpen értelmezzük:

Jó tudni: A jogosultság egy “**–**” jellel indul, ez jelenti azt, hogy ez egy egyszerű fájl. Amennyiben ez a fájl könyvtár *(hiszen az is fájl)*, úgy az első karakter egy “**d**” betű lenne. :)

Ezt az utolsó 9 karaktert “**rw-r–r–**” 3 részre oszthatjuk fel, pontosabban:

- a fájl tulajdonosa: **rw-**
- azok a felhasználók, akik abba a csoportba tartoznak amiben a fájl is van: **r–**
- és azok a felhasználók, akik se nem tulajdonosai a fájlnak és se nem tartoznak abba a csoportba amibe a fájl tartozik: **r–**

Láthatjuk, hogy

- a fájl tulajdonosa (**rw-**) olvashatja, írhatja, de nem futtathatja a fájl
- a fájl csoportjába tartozó felhasználók (**r–**) csak olvashatják a fájlt, írni és futtatni már nincs joguk
- az összes többi felhasználó (**r–**) akik nem a fájl tulajdonosai és nem is tartoznak a fájl csoportjába, azok is szintén csak olvashatják a fájlt

Nézzük meg ezt a táblázatot kicsit másképp, azaz most számokkal:

Jogosultság – ***chmod*** *parancs*			
Engedélyek	Tulajdonos	Csoport	Mindenki más
Olvasás	4	4	4
Írás	2	2	2
Futtatás	1	1	1
Összegezve:	7	7	7

Most nézzük meg, hogy majdnem ugyanaz mint az előző táblázat, csak annyival másabb, hogy itt már nem betűkkel van jelölve a jogosultság, hanem számmal.
Nézzük meg megint, hogy ha egy állománynak **644** a jogosultsága, akkor az mit is takarhat.
Most meg fogunk lepődni, hiszen ha ránézünk a két táblázatra, ez ugyanazt jelenti mint az előbb említett **rw-r–r–**, azaz:

- a fájl tulajdonosa (4+2=**6**) olvashatja, írhatja, de nem futtathatja a fájl
- a fájl csoportjába tartozó felhasználók (4+0=**4**) csak olvashatják a fájlt, írni és futtatni már nincs joguk
- az összes többi felhasználó (4+0=**4**) akik nem a fájl tulajdonosai és nem is tartoznak a fájl csoportjába, azok is szintén csak olvashatják a fájlt

A jogosultságokat a “**chmod**” paranccsal tudjuk elvégezni.

A következőképpen tudjuk módosítani a fájlunk (pl.: **gyakorlas.txt**) jogosultságait:

chmod

- **+** # Hozzáad egy engedélyt.
- **–** # Elvesz egy engedélyt.
- **=** # Beállítja az engedélyt.
- **r** # Olvasási engedély hozzáadása.
- **w** # Írási engedély hozzáadása.
- **x** # Végrehajtási / Futtatási engedély hozzáadása.
- **u** # Engedélyek beállítása a fájl, könyvtár tulajdonosának.
- **g** # Engedélyek beállítása a csoport számára.
- **o** # Engedélyek beállítása mindenki más számára.
- **a** # Engedélyek beállítása minden felhasználó számára. (tulaj, csoport, mindenki más)
- **-R** # A fájlok jogosultságait az alkönyvtárban is módosítja (rekurzív módon).

Példák:

`chmod u+x gyakorlas.txt` # Futtatási jogosultságot ad a fájl tulajdonosának.

`chmod go-rx gyakorlas.txt` # Visszavonja az olvasási és futtatási jogosultságot a csoport tagjaitól és mindenki mástól.

`chmod a=r gyakorlas.txt` # A fájl jogosultságait csak olvashatóra állítja minden felhasználó számára.

`chmod 444 gyakorlas.txt` # A fájl jogosultságait csak olvashatóra állítja minden felhasználó számára.

A fájl tulajdonosának és csoportjának megváltoztatására alkalmas parancsok:

`chown` # A fájl tulajdonosát változtatja meg.

- **-c** # Azon állományok nevét jeleníti meg, melyeknek a tulajdonosa megváltozott.
- **-f** # Tiltja a hibaüzenetek megjelenítését.
- **-R** # A fájlok tulajdonosát az alkönyvtárakban is módosítja.
- **-v** # A módosításokról részletes listát készít.

Példák:

`chown zsozso gyakorlas.txt` # A fájl “**zsozso**” tulajdonába kerül.

`chgrp felhasznalo gyakorlas.txt` # A fájl a “**felhasznalo**” csoportba kerül.

`chown zsozso:felhasznalo gyakorlas.txt` # A fájl “**zsozso**” tulajdonába és a “**felhasznalo**” csoportba kerül.

Azt hiszem ennyi lenne a jogosultságokról szóló rész. :)

Linux - Alapparancsok

Ls listázza a könyvtárat
ls -l állományok listázása
drwxr-xr-x 2 okj14f okj14f 4096 szept 21 17:25 Asztal
rwxr – felhasználói jogok
-xr-x – másokra való jogok
2 – hány helyről érhető el ez a file
okj14f okj14f - felhasználó és a csoport, amelybe tartozik
4096 – 4kb-os könyvtár bejegyzés
szept 21 17:25 – létrehozás dátuma
Asztal – állomány neve
ls -l -a rejtett állományok megjelenítése listázva
ls -h fájlméret emberi formátumban
ls -H ez 1000-as váltószámot használ
ls --help segítség kékrés
. jelenlegi mappa
.. szülő mappa ezeket a -a elhagyja
ls -r rendezés valami sorrend szerint
man leírás a kereset parancsról
pl man ls kilépés a q-val
df szabad hely a lemezen
df -h
who milyen felhasználóként vagyok ejelentkezve
who -a
whoami milxen felhasználóval vagyok bejelentkezve
top feladatkezelő, folyamatosan frissülő statisztika
cd mappanév/ belépés a mappába
ls -l -a
mkdir mappanév mappa létrehozása
mkdir 14{a,b,c,d,e} sok mappa létrefozása, amelyeknek a végei különböznek
mkdir -p iskola/pataky/okj/esti/14f -p vel létrehozom a szülő könyvtárakat, majd ezekben létrehozza a megfellő mappákat
ls -R mappa struktúra
rmdir mappanév mappa törlése
rmdir -p iskola/pataky/okj/esti/14f a teljes mappa lánc törlése
viszont, ha ebben van elágazás, akkkor csak az elágazásig törli le
rmdir 14{a,b,c,d,e} a változó végü mapák törlése
mkdir {9,10,11,12}{a,b,c,d,e,f} mindent létrhoz
mkdir -p dolgozat/{9,10,11,12}{a,b,c,d,e,f}
rmdir -p dolgozat/{9,10,11,12}{a,b,c,d,e,f}
mkdir vitagh\ krisztian a létrehozott mappa nevében legyen szóköz

Jogosultság állítás
chmod -R

-R minden almappára is igaz legyen
u user
g group
o ohter
a all
- elveszem a tulajdonságot
+ hozzáadok tulajdonsgot
= egyenlővéteszem
chmod g-w iskola/ iskola mappa csoport irásjog elvétele
chmod a+w iskola/ mindenkinek adot írásjogot
chmod 750 iskola/ 7 user 5 group 0 other
chmod go+rwx iskola/ több tulajdonság egyben
ha lecserélem a csoportot, akkor csak other csoport tulajdonságok fognak rám is vonatkozni.

```
-rw-r--r-- 1 okj14f okj14f 8980 szept 21 17:24 examples.desktop
drwxrwxrwx 2 okj14f okj14f 4096 szept 21 21:04 iskola
drwxr-xr-x 2 okj14f okj14f 4096 szept 21 17:25 Képek
drwxr-xr-x 2 okj14f okj14f 4096 szept 21 17:25 Letöltések
```

Zölddel kiemelt mappa, amelyet mindenki futtathat.
chown rgazda iskola/ tulajdonos megváltoztatása
chgrp rgazda iskola/ csoport megváltoztatása
cd / -> gyökér mappa
cd ~ -> home mappa
du -> lemez használat
du Dokumntumok -> dokumentumok mappa lemezhasználata
less .bash_history -> kiírja a .bash_history tartalmát a kijelzőre
grep -> minta keresése
ls -l | grep "^d" | "r-xr-xr-x" -> listázza a azokat a mappákat amik csak amiknél csak olvasási és futtatási jog van (a **^** kalap a sor elejét nézi a **$** dollár jel pedig a sor végét)
touch mondat.txt -> létrehozza a mondat.txt fájlt
nano mondat.txt -> nano szövegszerkesztő (**ctrl+x** = bezárás, **shift+ctrl+v** = beillesztés)
cat mondat.txt -> kiírja a mondat.txt tartalmát (az első sorral kezdi)
tac mondat.txt -> kiírja a mondat.txt tartalmát (az utolsó sorral kezdi)
cat .bash_history | grep "^ls" | wc -> a wc (word counter) megszámolja a ls-el kezdődő sorokat (3 kimenete van)
head -2 mondat.txt -> kiírja az első két sort a mondat.txt-ből
tail -2 mondat.txt -> kiírja az utolsó két sort a mondat.txt-ből
sort mondat.txt -> sorba teszi a sorokat abc szerint és kiírja a képernyőre
ls -lR | grep ".txt$" -> kilistázza az összes .txt-re végződő fájlt
more mondat.txt -> meg lehet vele nézni a fájlt (tovább -> **space** = oldalanként, **enter** = soronként)
sync -> kiírja adathordozóra a memória tartalmát (pl. pendrive

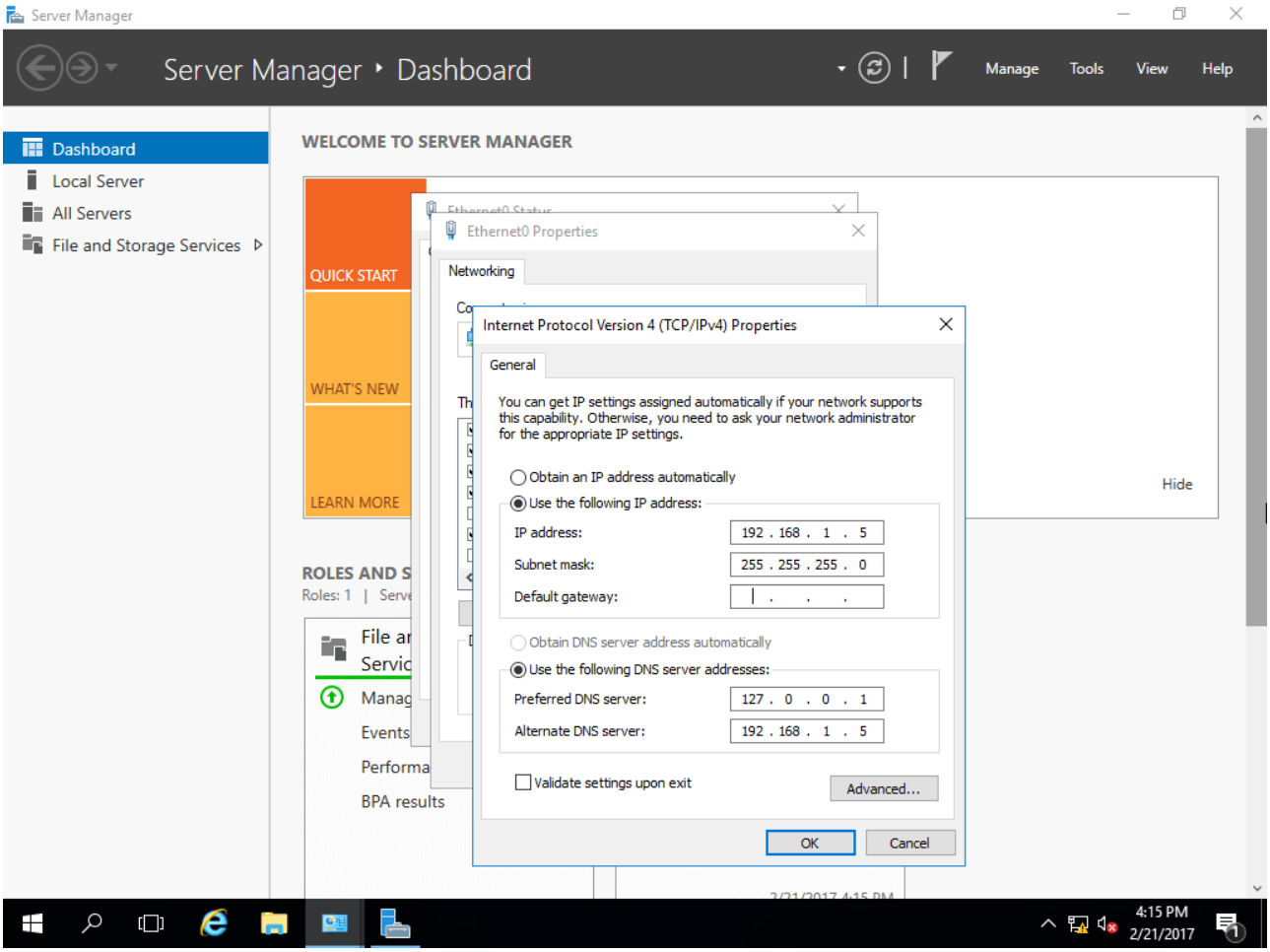
eltávolítása előtt érdemes lefuttatni)
mv mondat.txt mondat.doc -> mozgatás, átnevezés
cp mondat.txt mondat.doc -> másolás
rm mondat.doc -> törlés
ln -> linkelés (hard link, ugyan arra a lemezterületre mutat),
ln -s -> (soft link = windows parancsikon)
ctrl+l -> megnyomása letakarítja a kijelzőt = **clear** parancs
ps -> processzek (kapcsolók **-a -A -u**)
pwd -> aktuáluis mappa kilistázása
date -> rendszeridő (**-s** beállítás)
find -> keresés
find > valami.list -> beleirányitja a kimenetet
find | grep ".jpg" > kepek.list -> kigyűjti a jpg fájlokat a kepek.list fájlba
find | grep ".jpeg" >> kepek.list -> hozzácsatolja az eredményt
file monda.txt -> megmondja a fájl típusát
cmp mondat.txt mondat.doc -> fájlokat hasonlít össze
rev mondat.txt -> kiírja a mondat.txt tartalmát jobbról balra

Windows server 2016

Active Directory telepítése

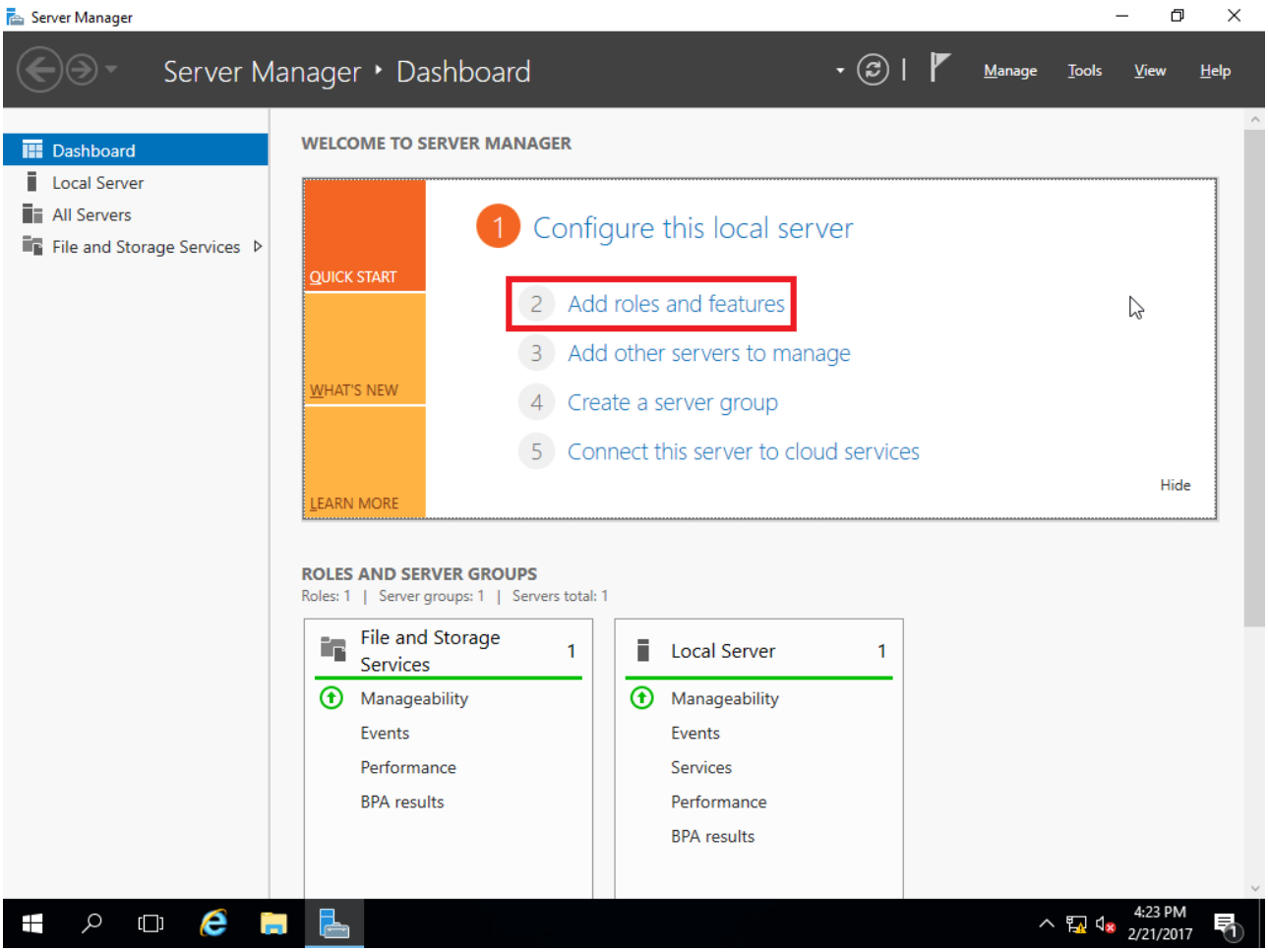
Néhány előkészület:

- Hozzon létre erős jelszót az Adminisztrátori fiókhoz (a jövőben Domain Admin lesz);
- Telepítse az összes frissítést;
- Nevezze át a szervert a vállalati elnevezési irányelvek alapján. (A Windows Server telepítésekor véletlenszerű név keletkezik);
- Állítson be statikus IP-t a kiszolgálóra.

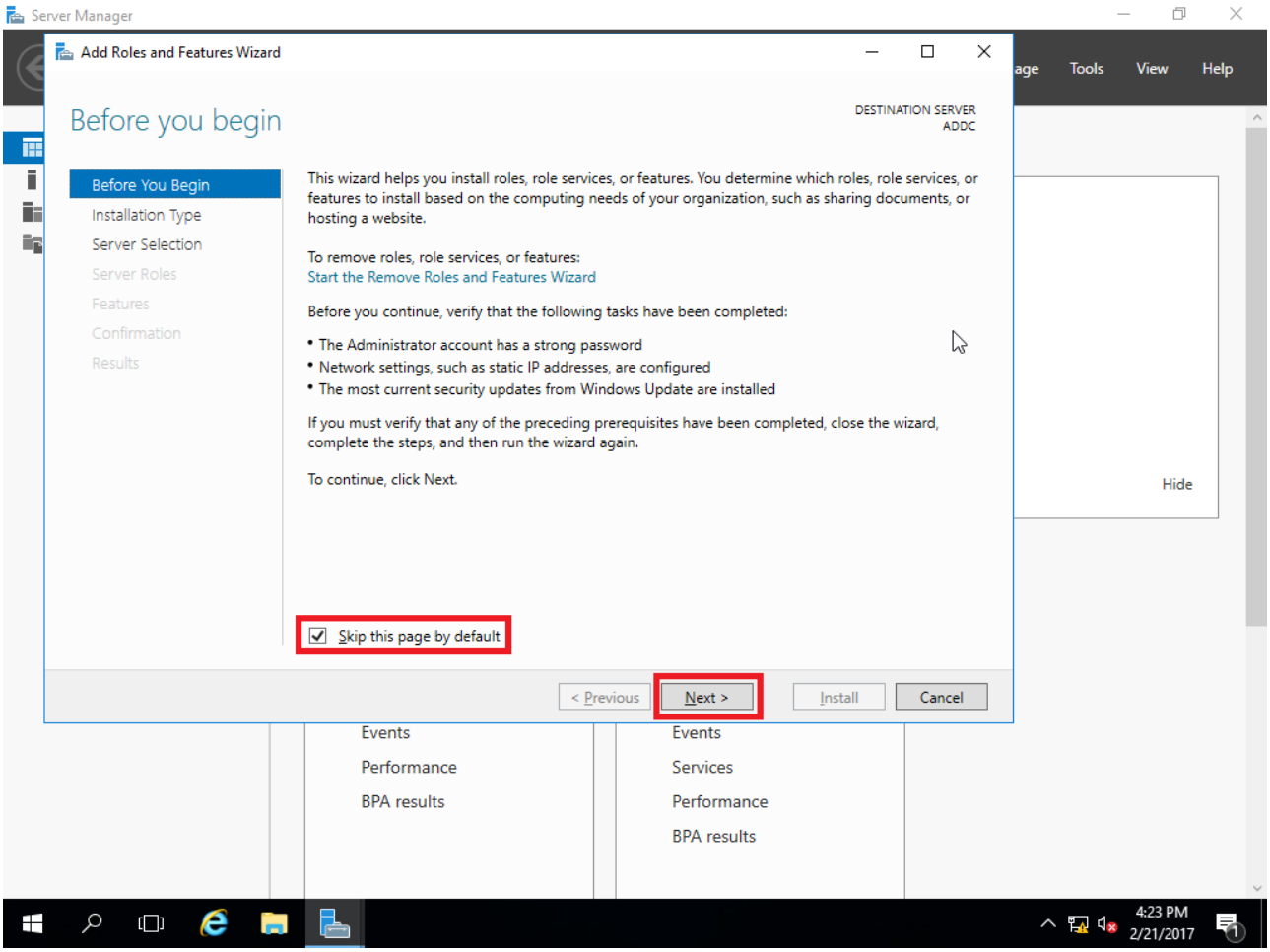


Telepítés:

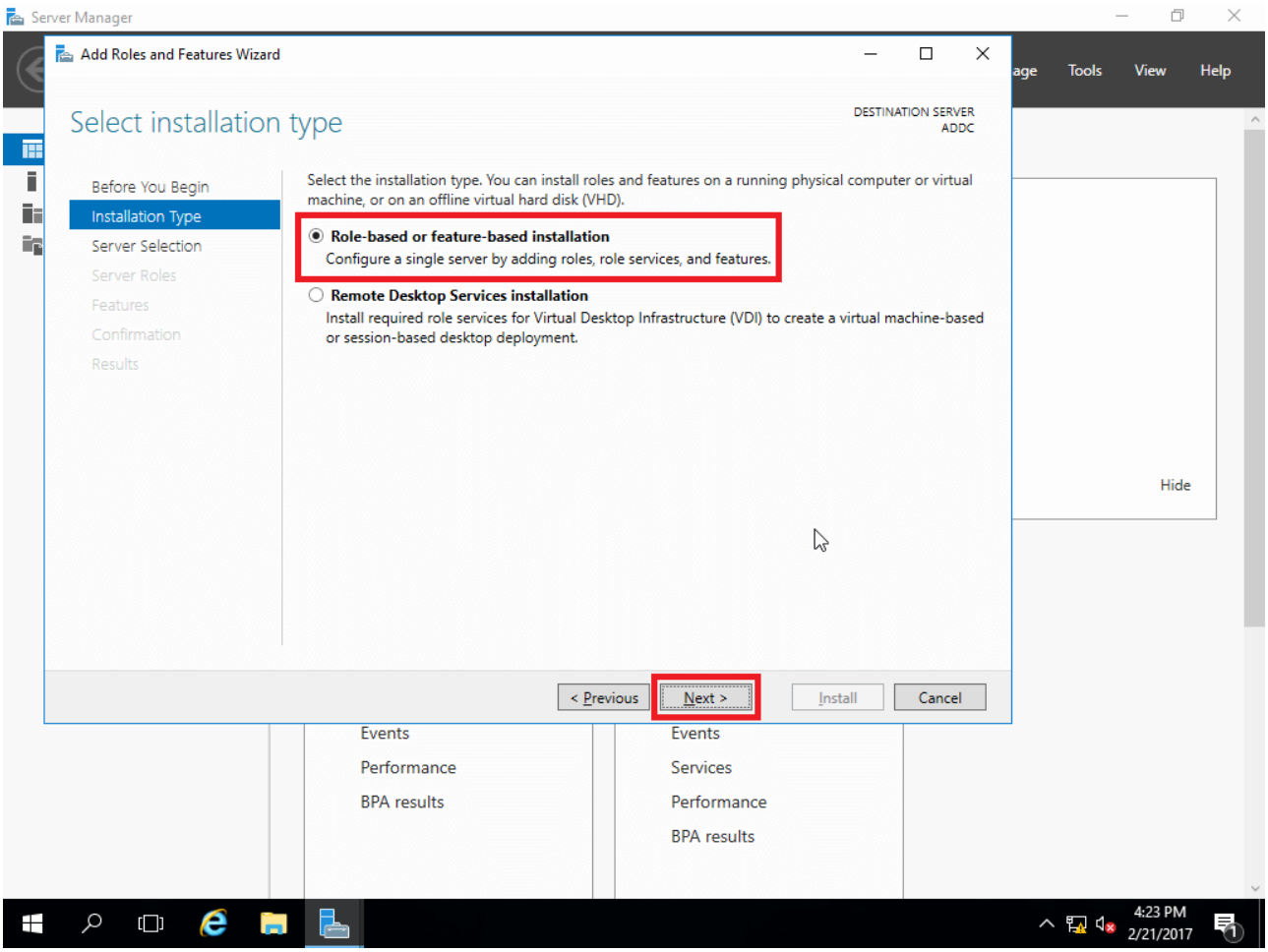
1. Miután elindult a Kiszolgálókezelő - kattintson a **Róluk és funkciók hozzáadása linkre**;



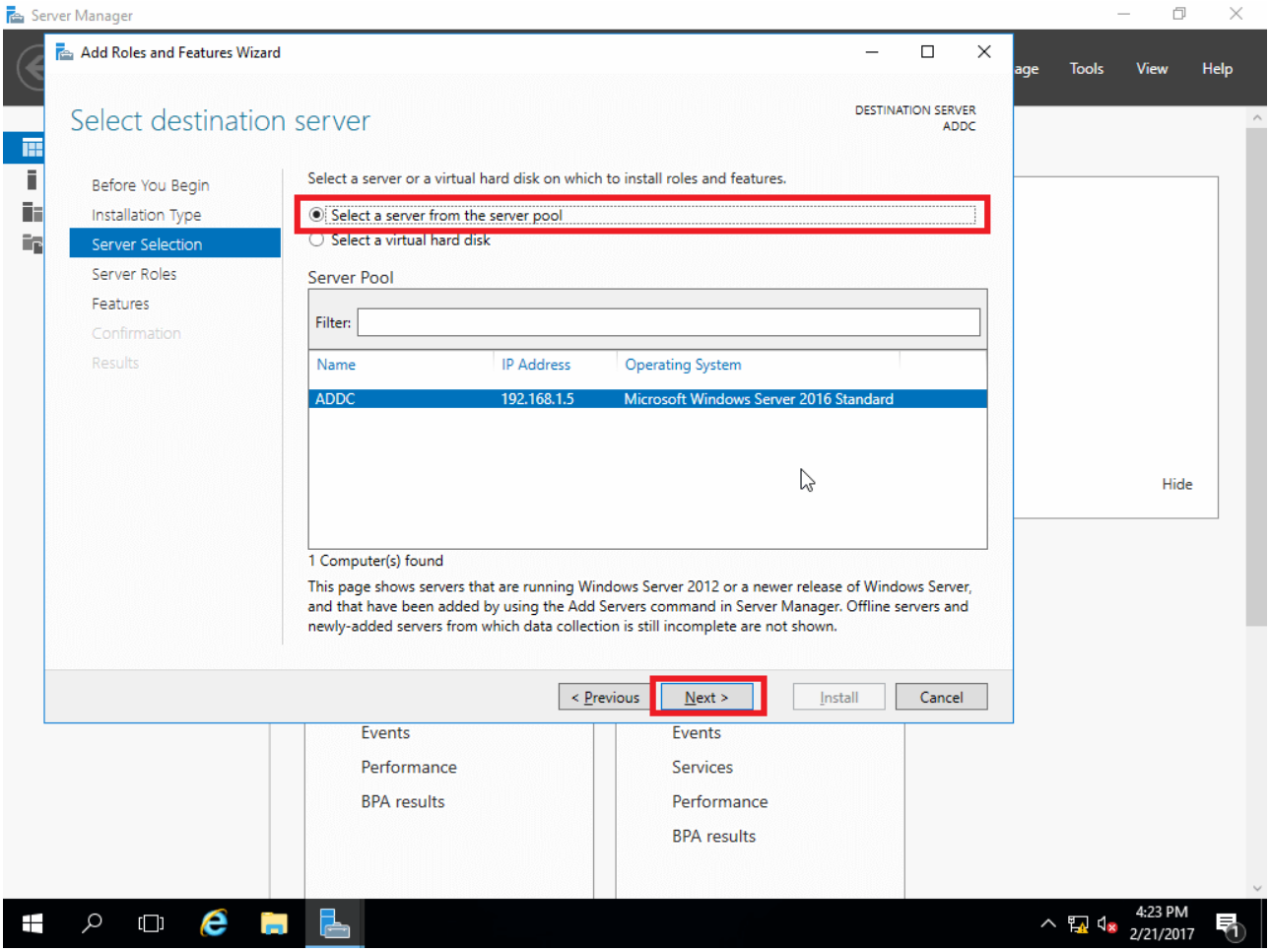
2. Kattintson a **Next** (**Következő)** gombra a varázsló képernyőjén. Ezenkívül beállíthatja **az alapértelmezés szerint az Oldal kihagyása** jelölőnégyzetet, és ezt a lépést a következő futásidőben nem látja;



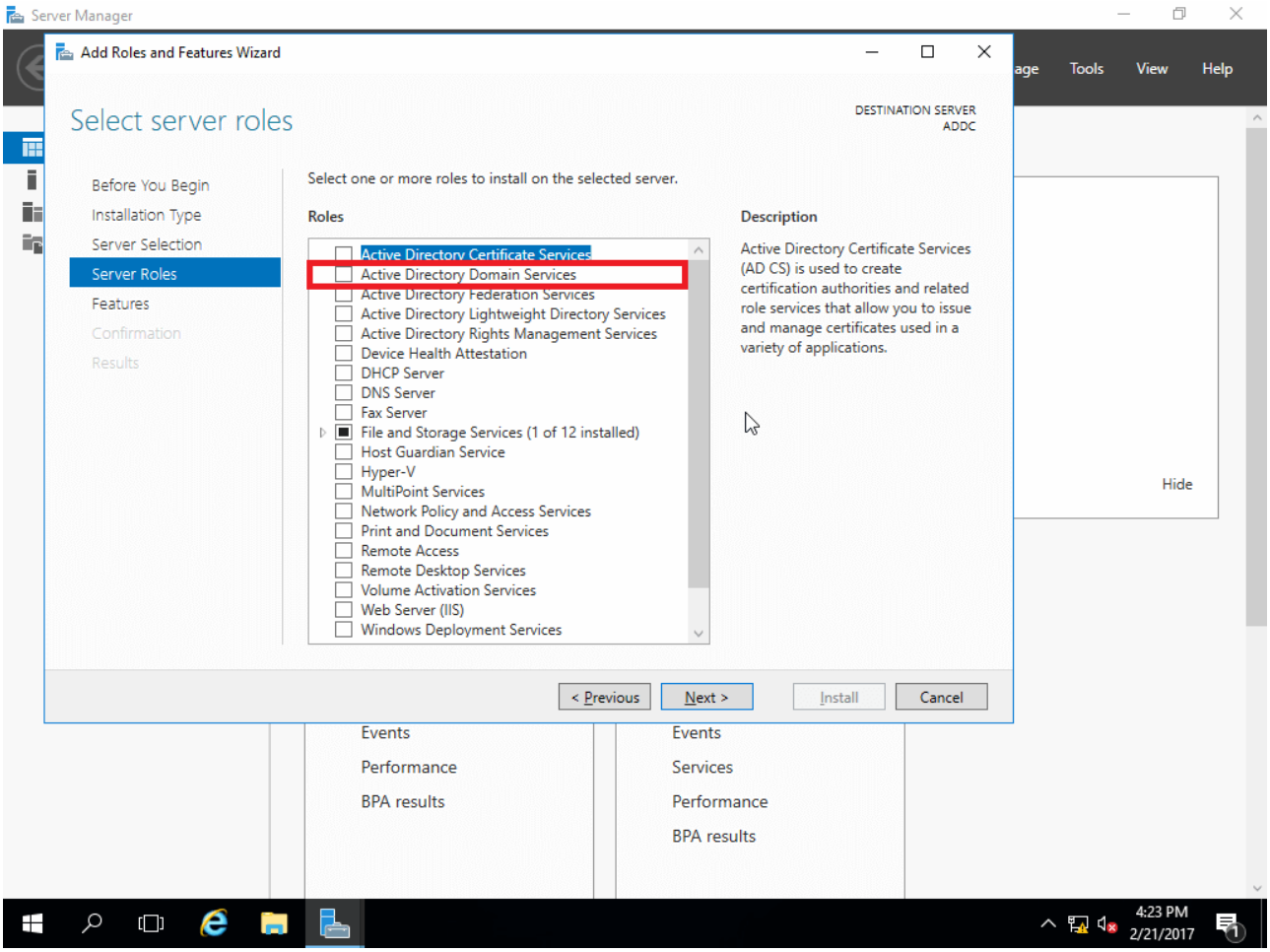
3. A telepítés típusához válassza a **Szerepkör alapú vagy funkcióalapú telepítést,** majd kattintson a **Tovább** gombra;



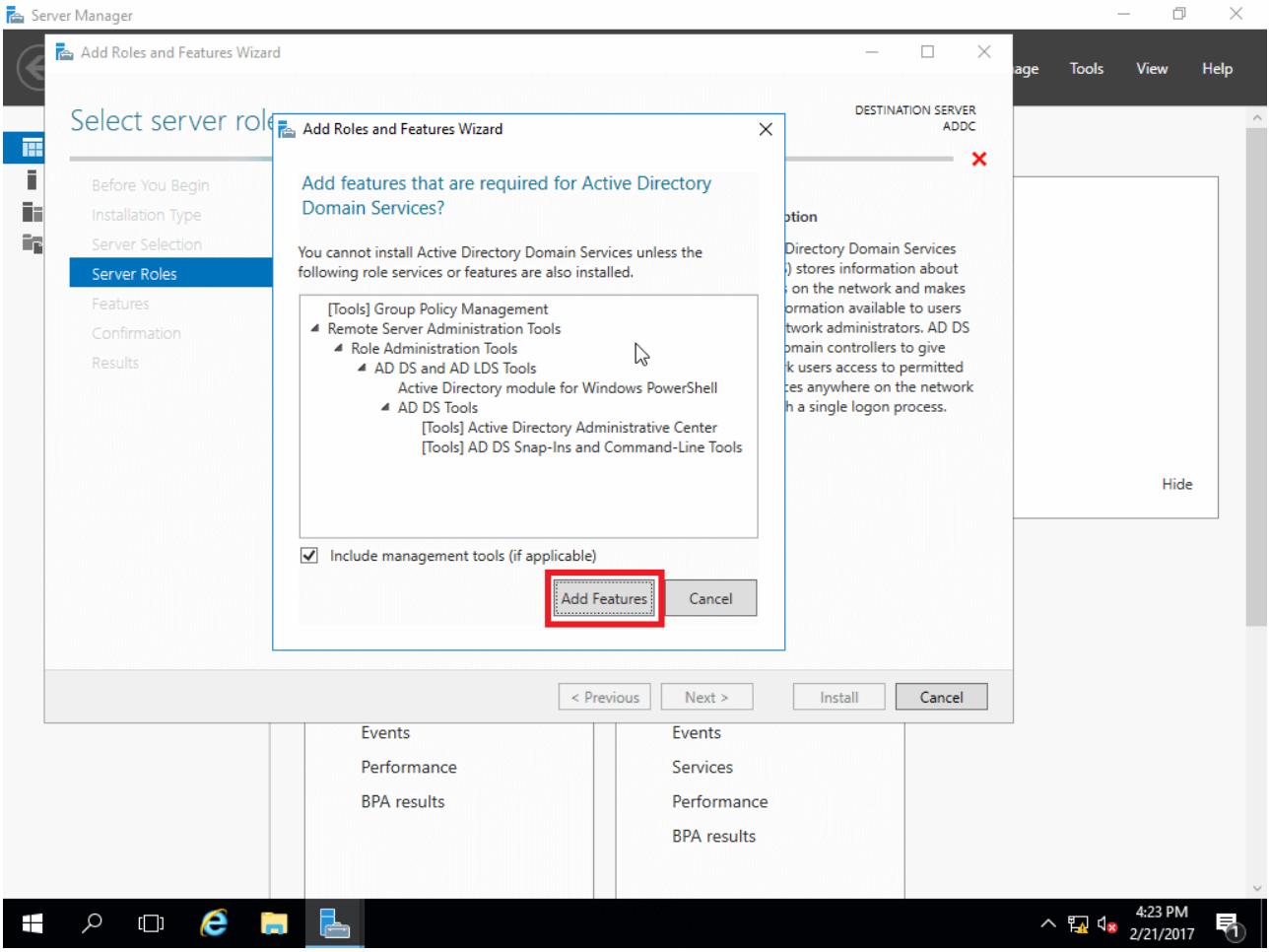
4. Válassza ki az opciót. Válasszon **ki egy kiszolgálót a kiszolgáló-gyűjteményből**(alapértelmezés szerint kiválasztva), majd kattintson a **Következő** gombra;



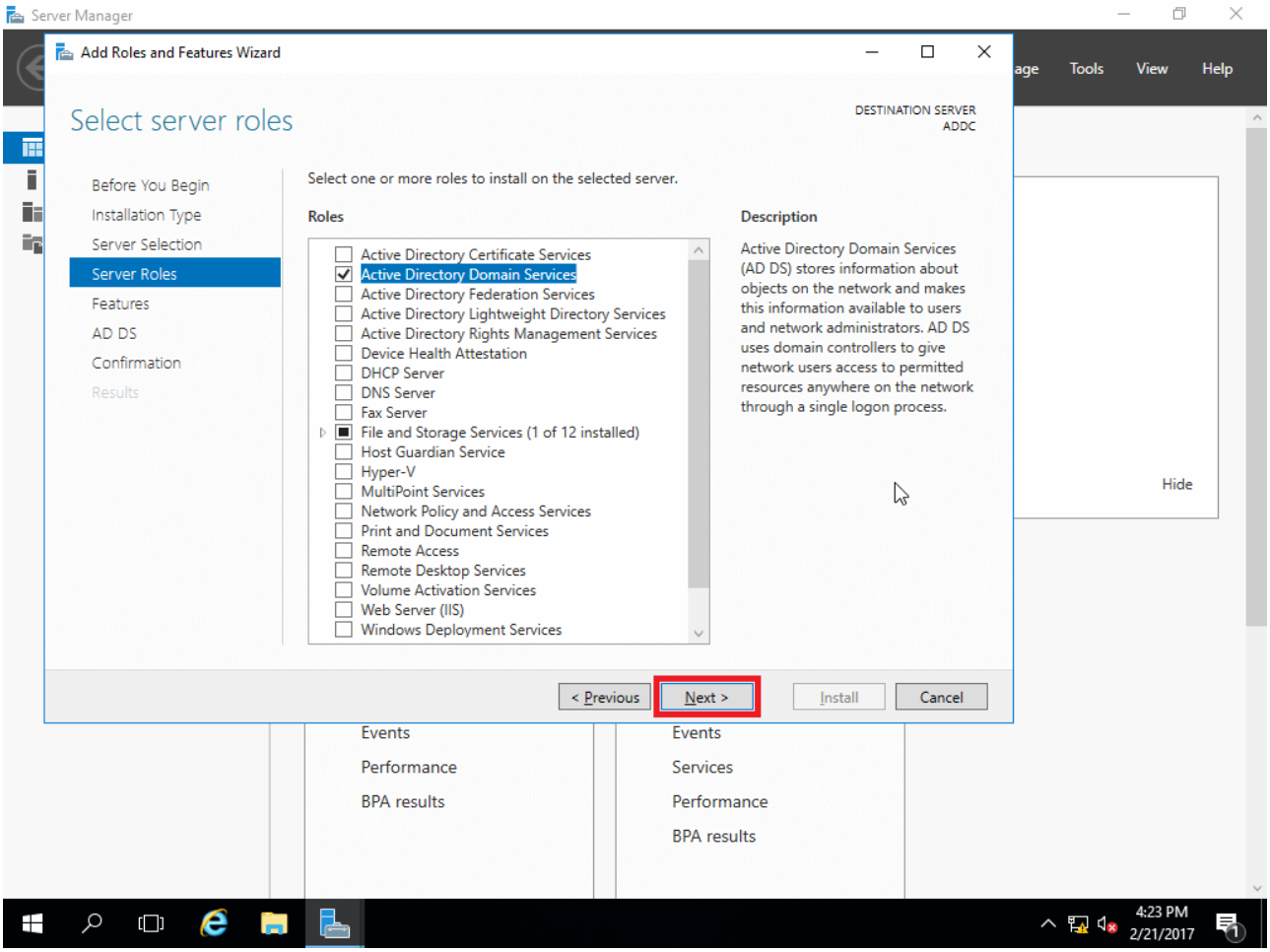
5. Kattintson az **Active Directory Domain Services szolgáltatáshoz** .



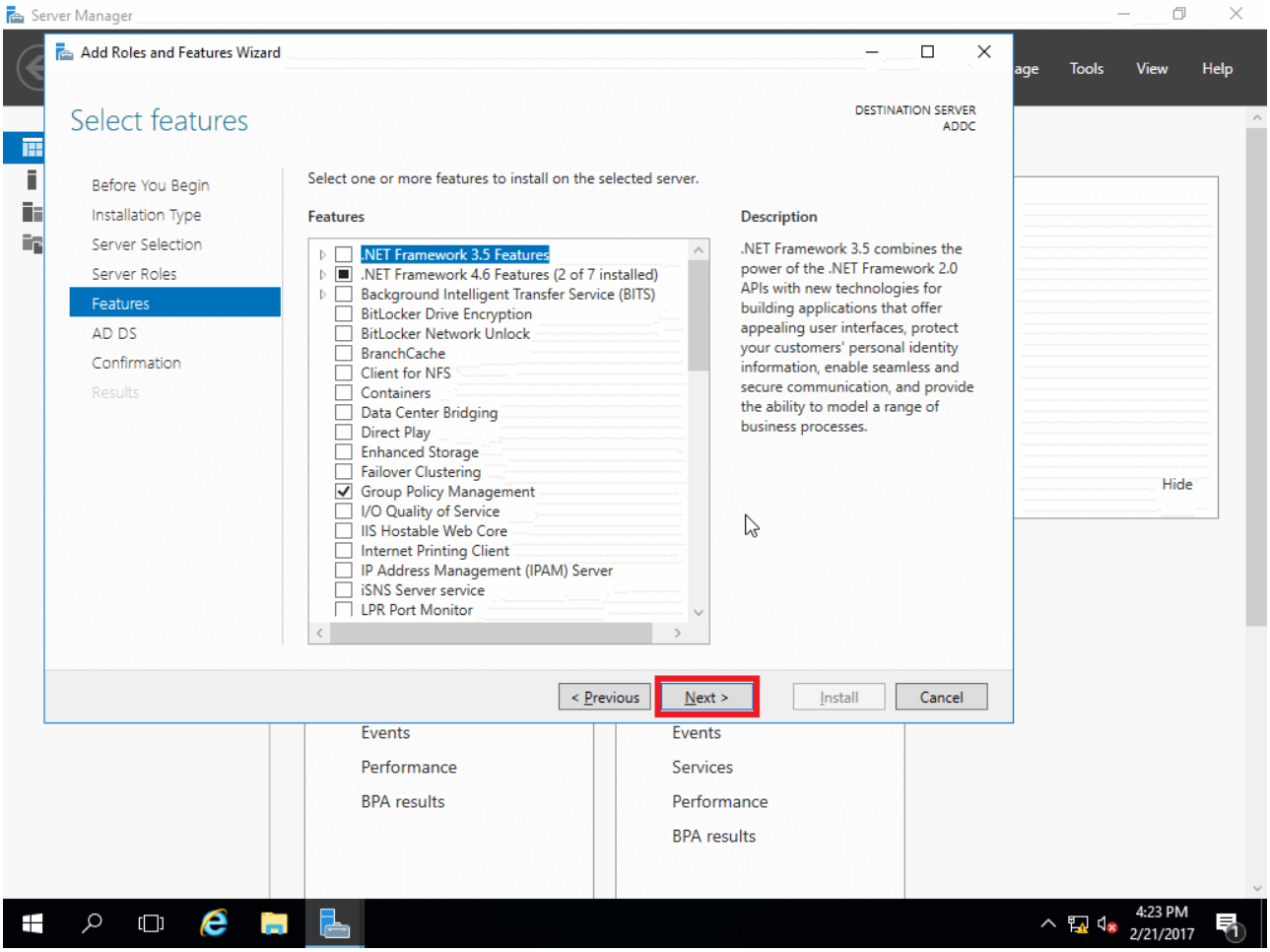
6. Az új felbukkanó ablakban kattintson a **Szolgáltatások hozzáadása** elemre ;



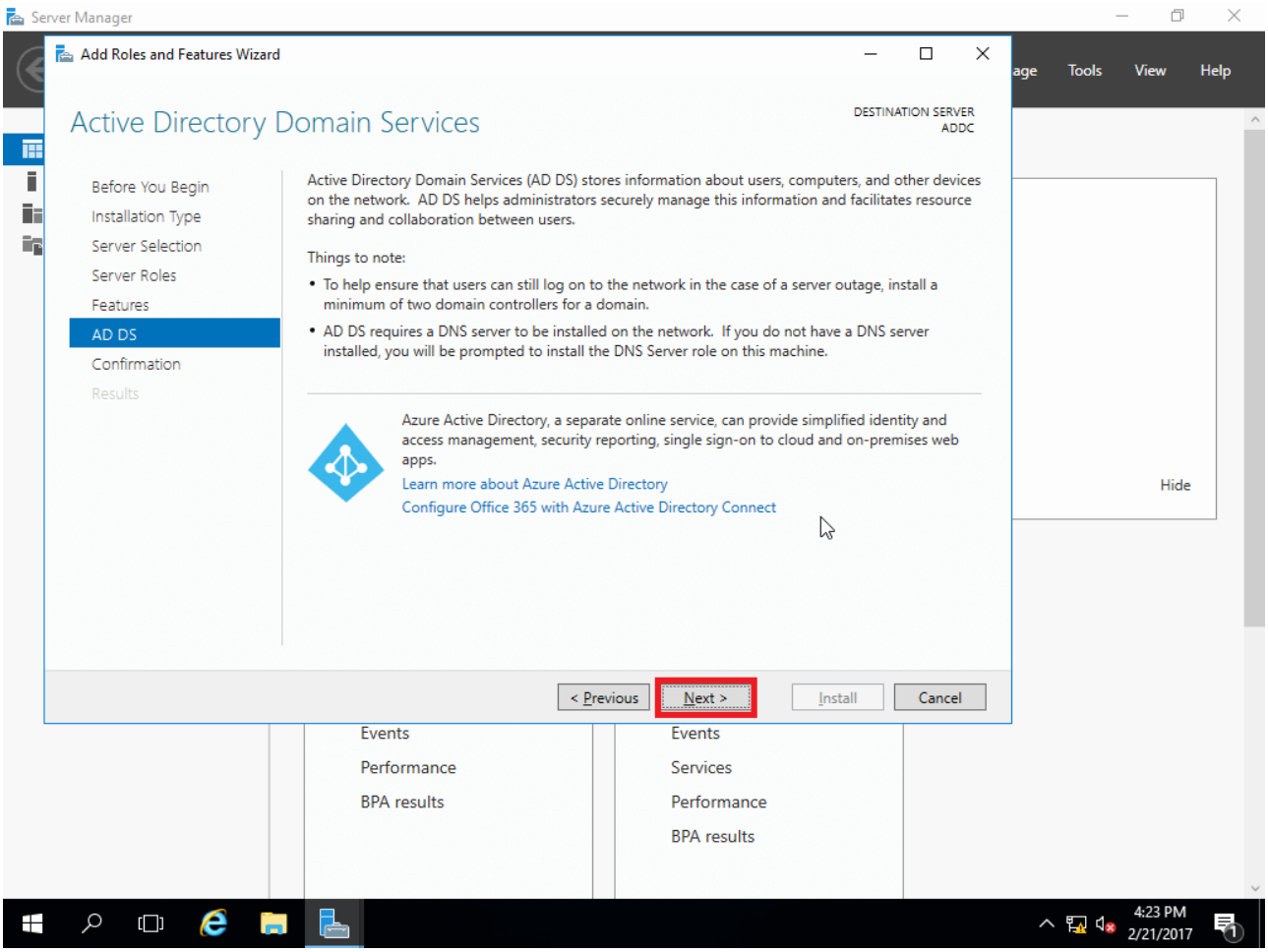
7. Kattintson a **Tovább** gombra;



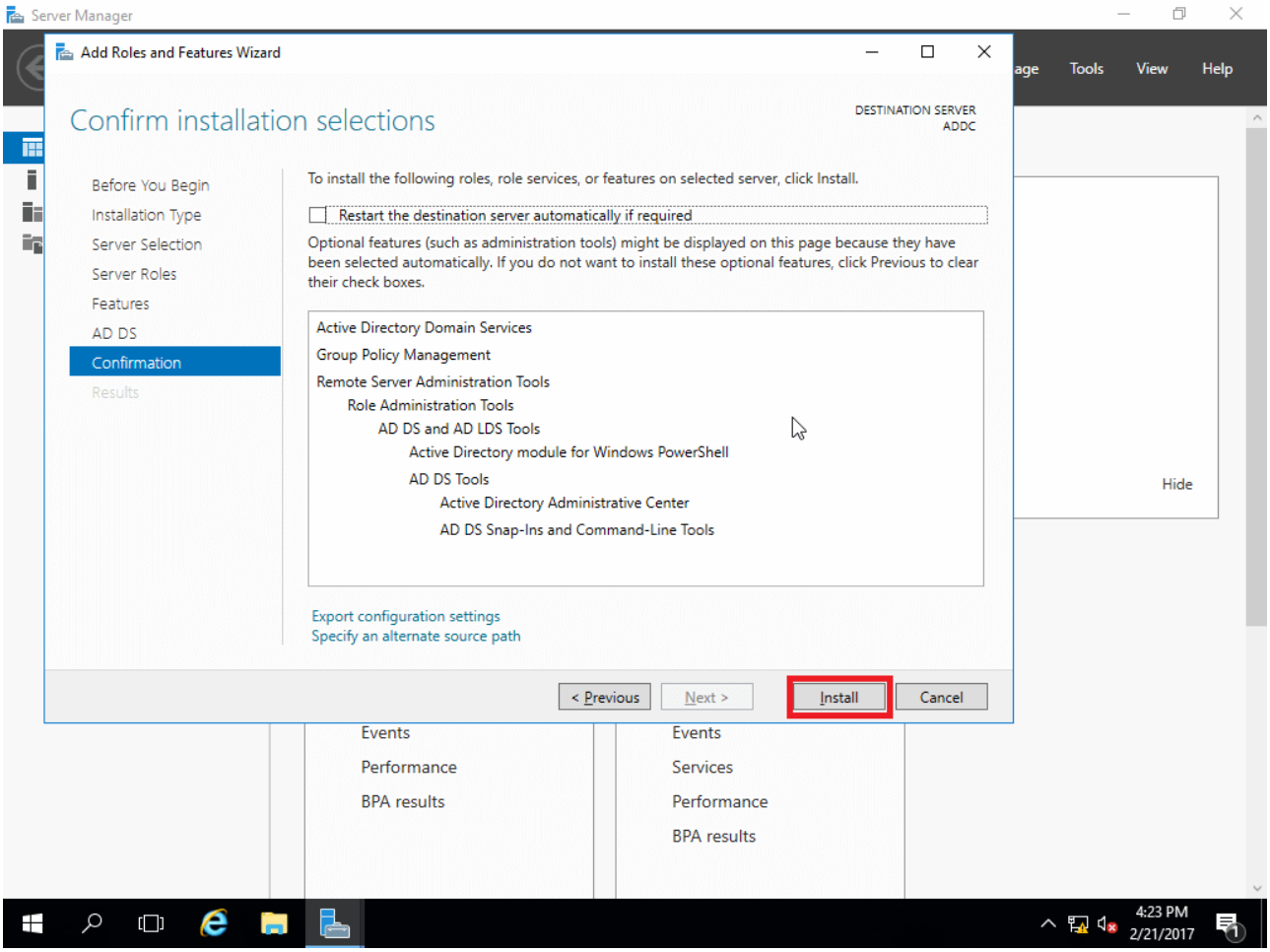
8. A funkciók esetében ne válasszon semmit, kattintson a **Következő** gombra;



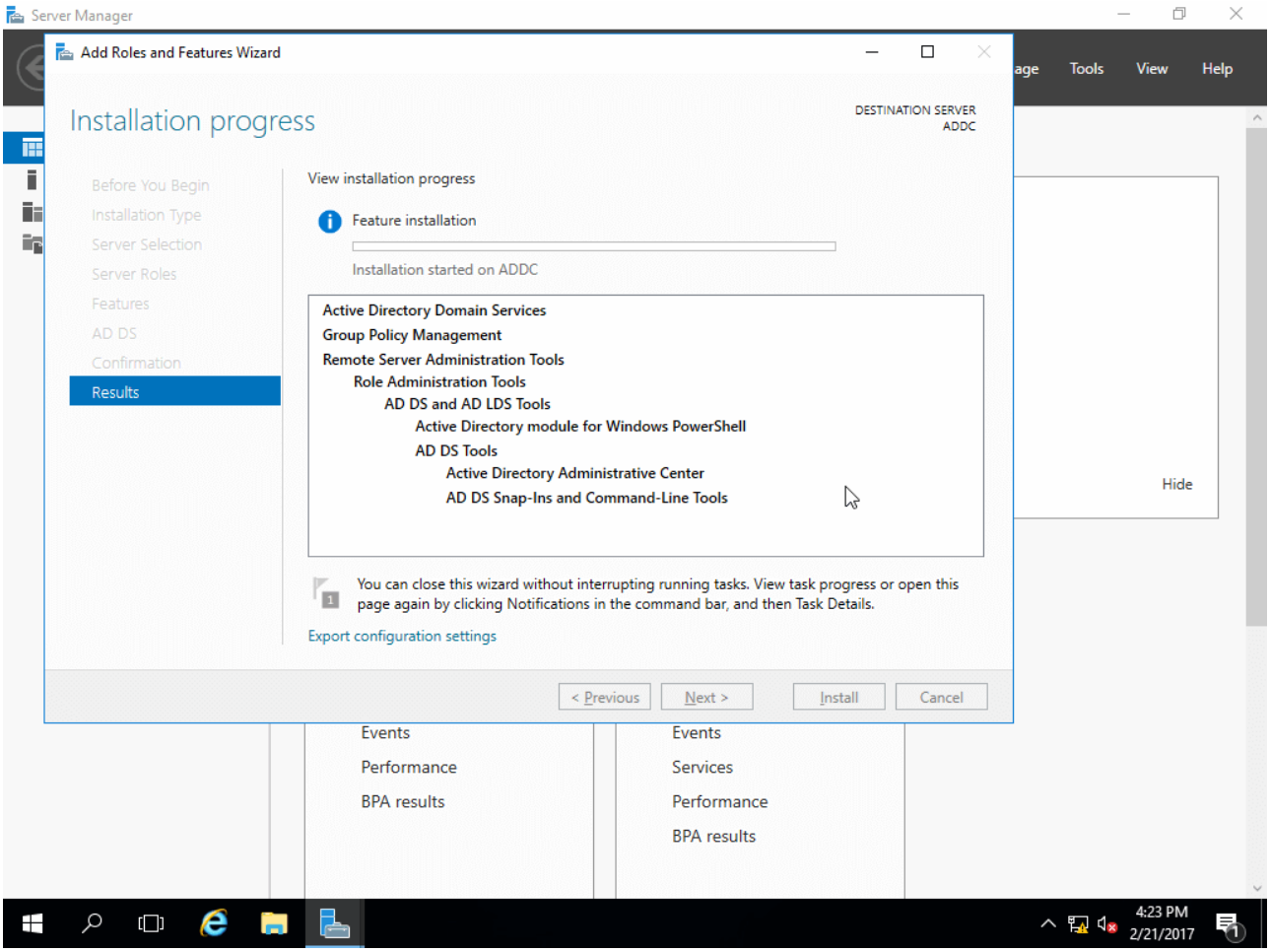
9. Az Active Directory tartományi szolgáltatásoknál kattintson a **Következő** gombra;



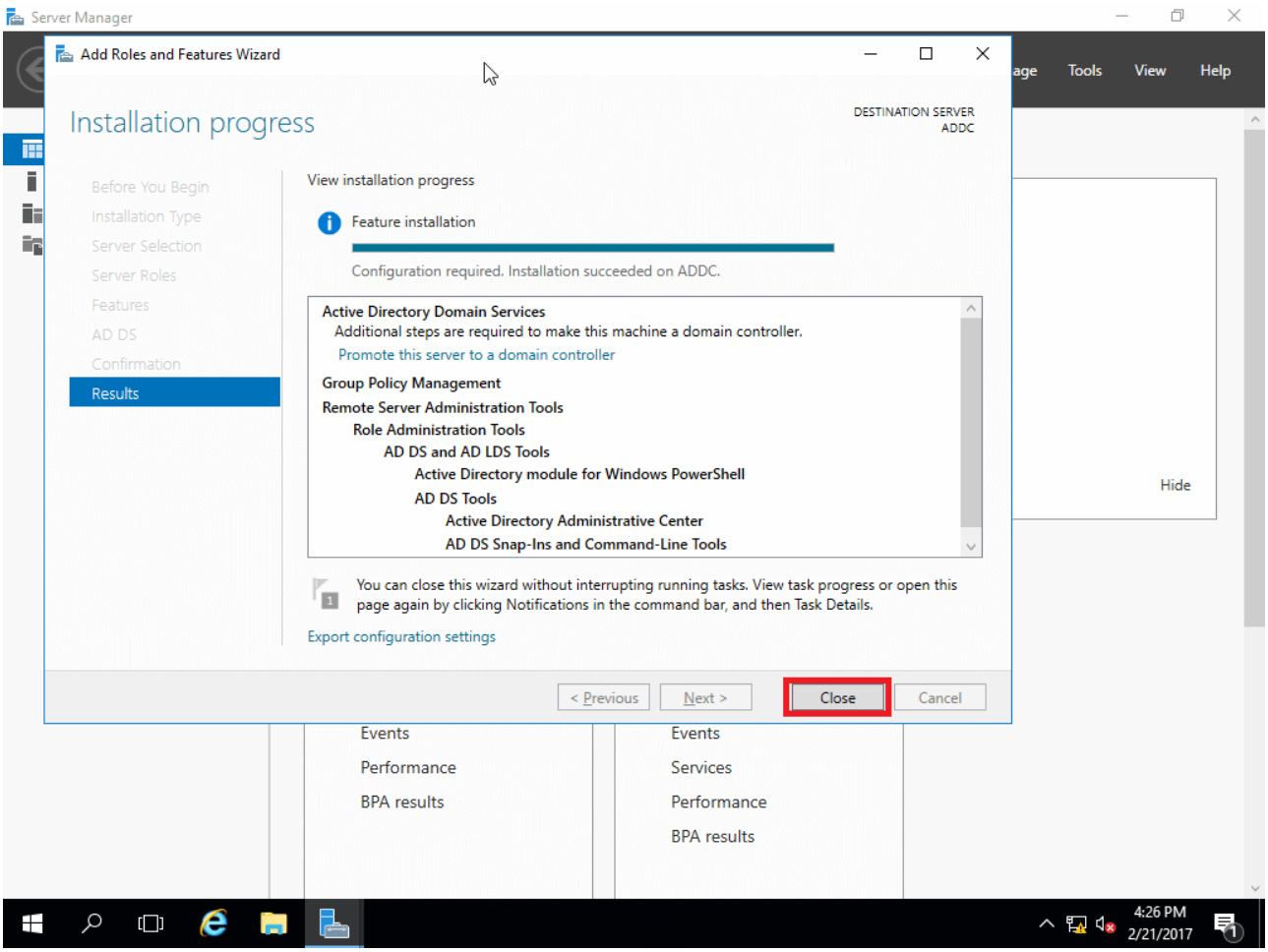
10. A visszaigazoló képernyőn eldöntheti, hogy újra kívánja-e indítani a célkiszolgálót, vagy nem, esetemben ezt az opciót nem állítom be. Kattintson a **Telepítés** gombra;



11. Telepítés kezdődik;

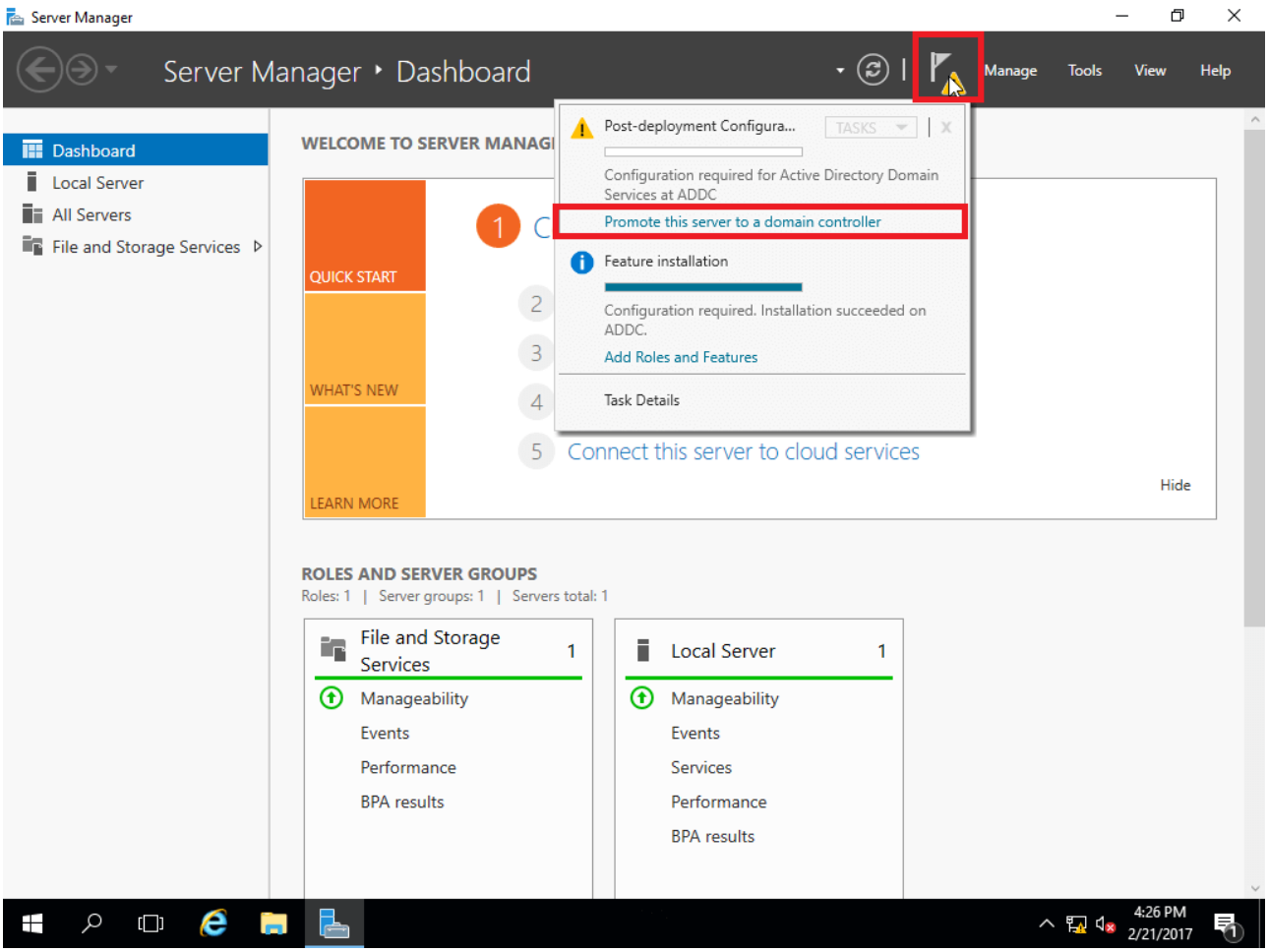


12. Miután befejeződött, kattintson a **Bezárás** gombra.

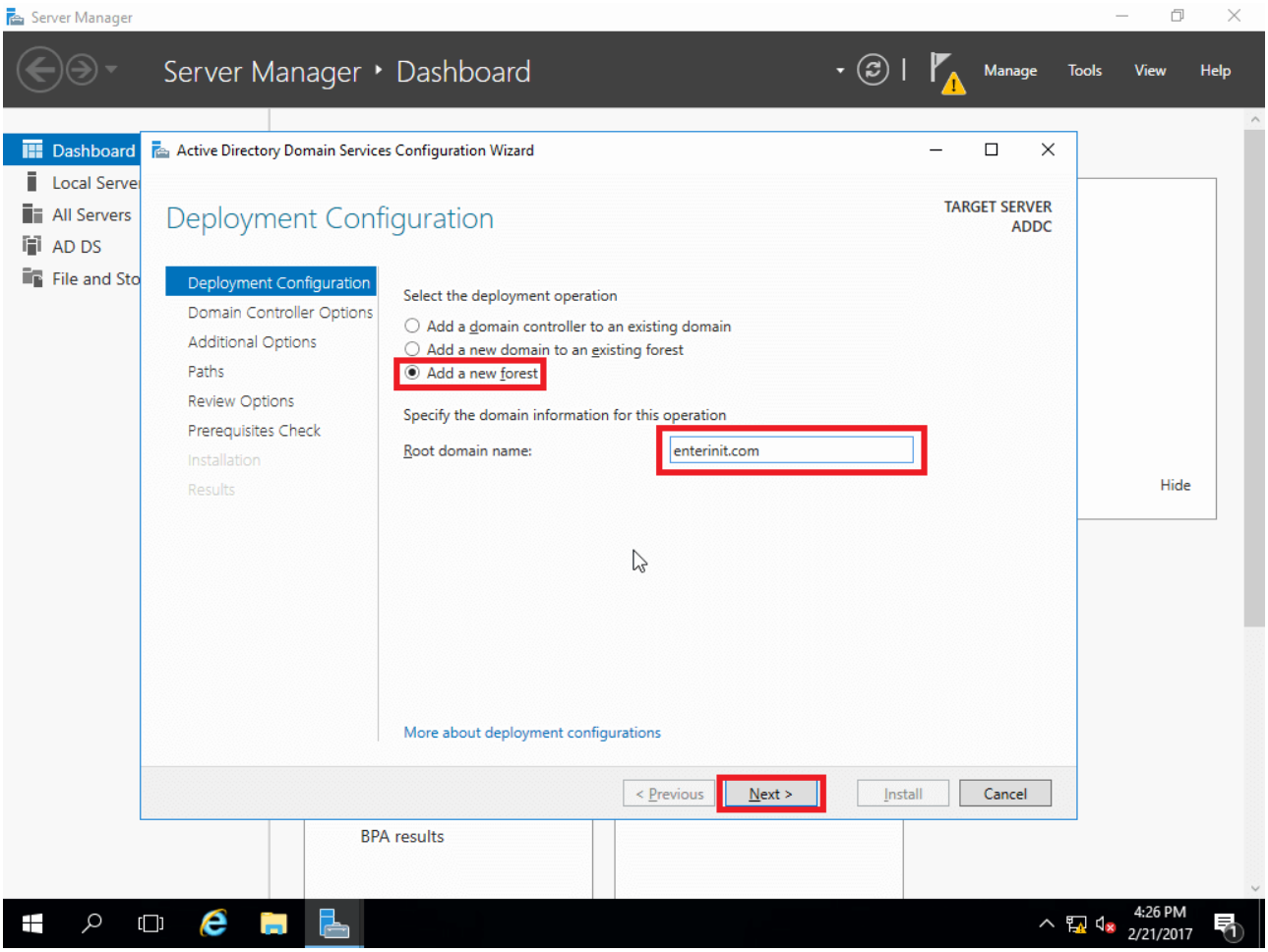


konfiguráció:

1. A **Kiszolgálókezelő irányítópultján** kattintson a felkiáltójel jelzéssel ellátott zászlóra. Kattintson **a kiszolgáló támogatása egy tartományvezérlőre** ;

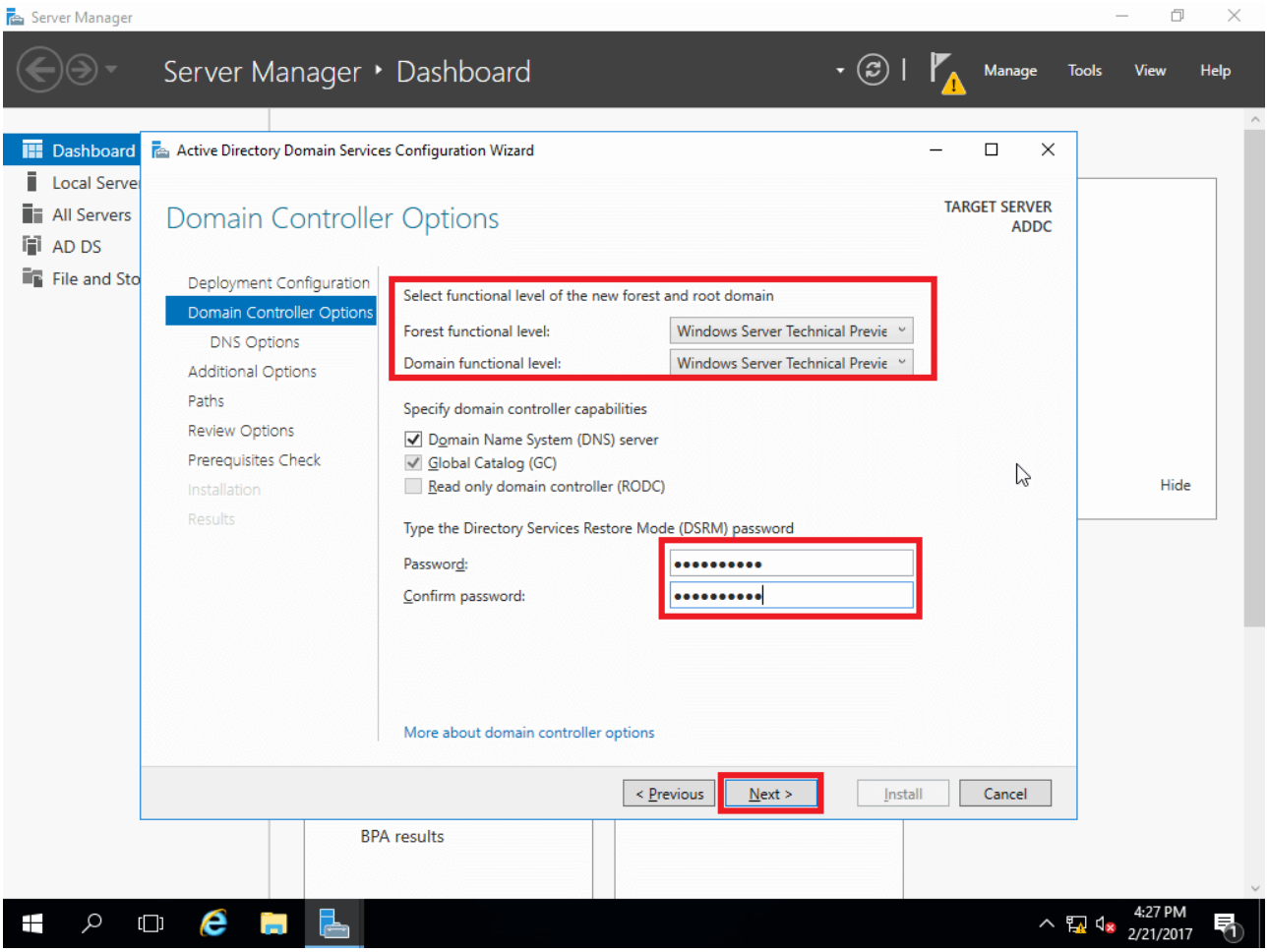


2. A következő képernyőn válassza ki a szükséges telepítési műveletet. Ebben a példában az **Új erdő hozzáadása** van kiválasztva, és a neve **enterinit . com be** van adva, hogy létrehozzon egy új gyökértartományt. Kattintson a **Következő** gombra;

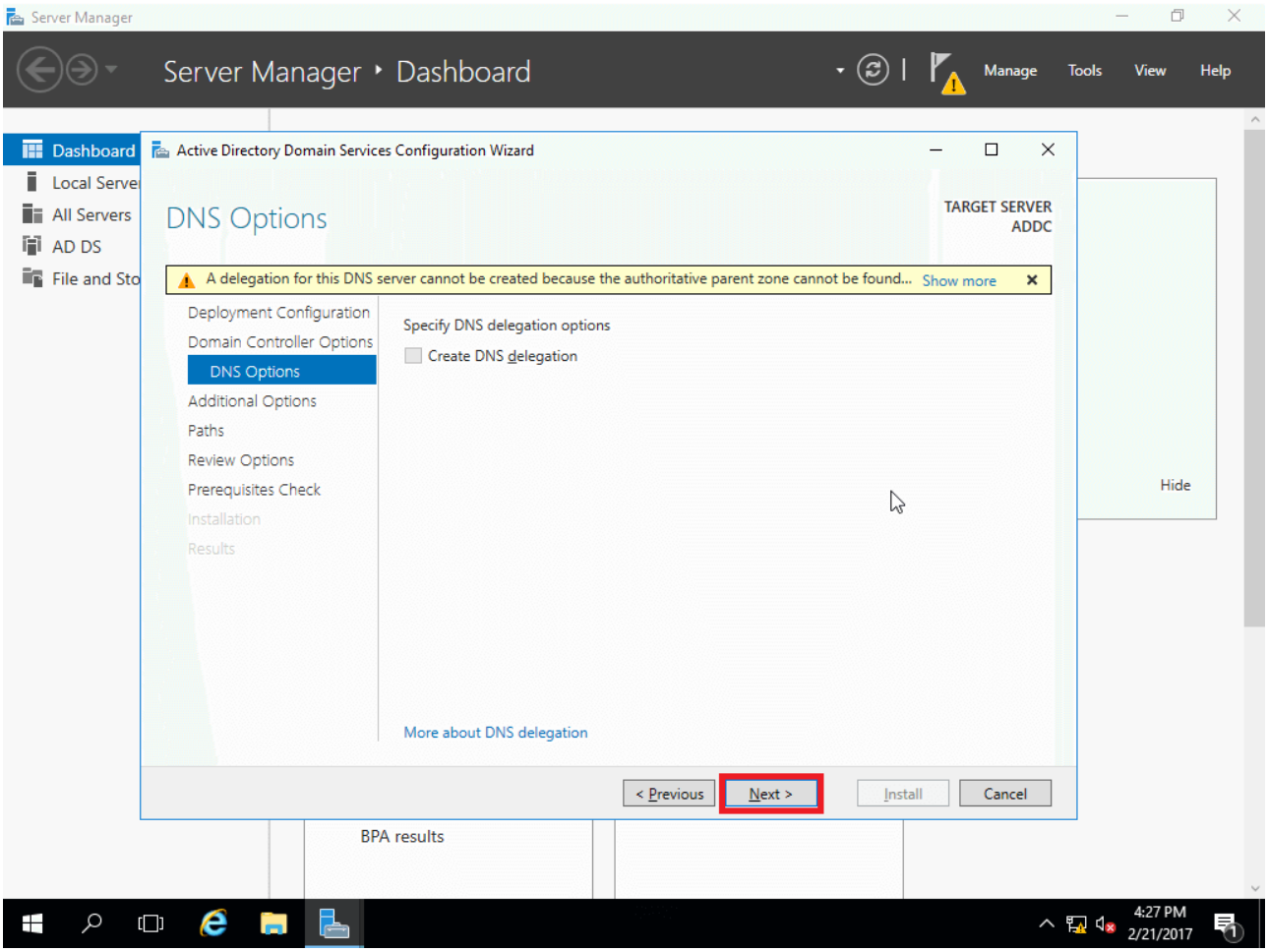


3. Mivel ez lesz az egyetlen tartományvezérlő ebben a laboratóriumi példában, mind az erdő, mind a tartományi funkcionális szintek a **Windows Server műszaki előnézetében**maradnak (a Windows Server 2016-ot a VLSC-ből, de technikai előnézetként ismerem). Jelölje be a **Domain Name System (DNS) kiszolgáló** jelölőnégyzetét , hogy ez a rendszer DNS-kiszolgáló legyen. A GC opciót anélkül ellenőrizzük, hogy módosíthatnánk, mivel az első tartományvezérlő **Globális Katalógus** kiszolgálónak kell lennie . A harmadik opció nincs bejelölve és nem módosítható, mivel az első tartományvezérlő nem lehet csak **olvasható tartományvezérlő** ;

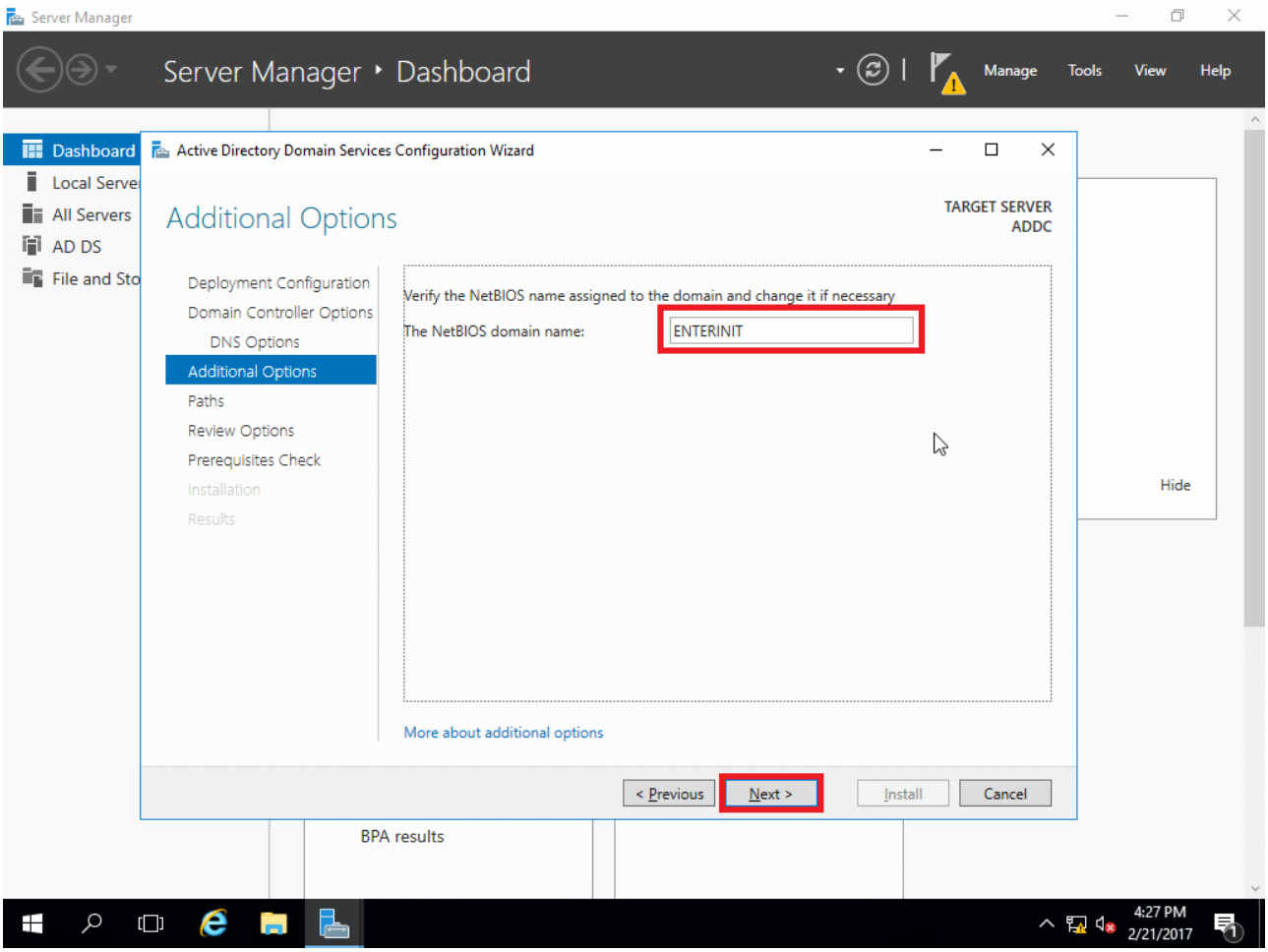
4. Adjon meg egy DSRM jelszót, majd kattintson a **Tovább** gombra .



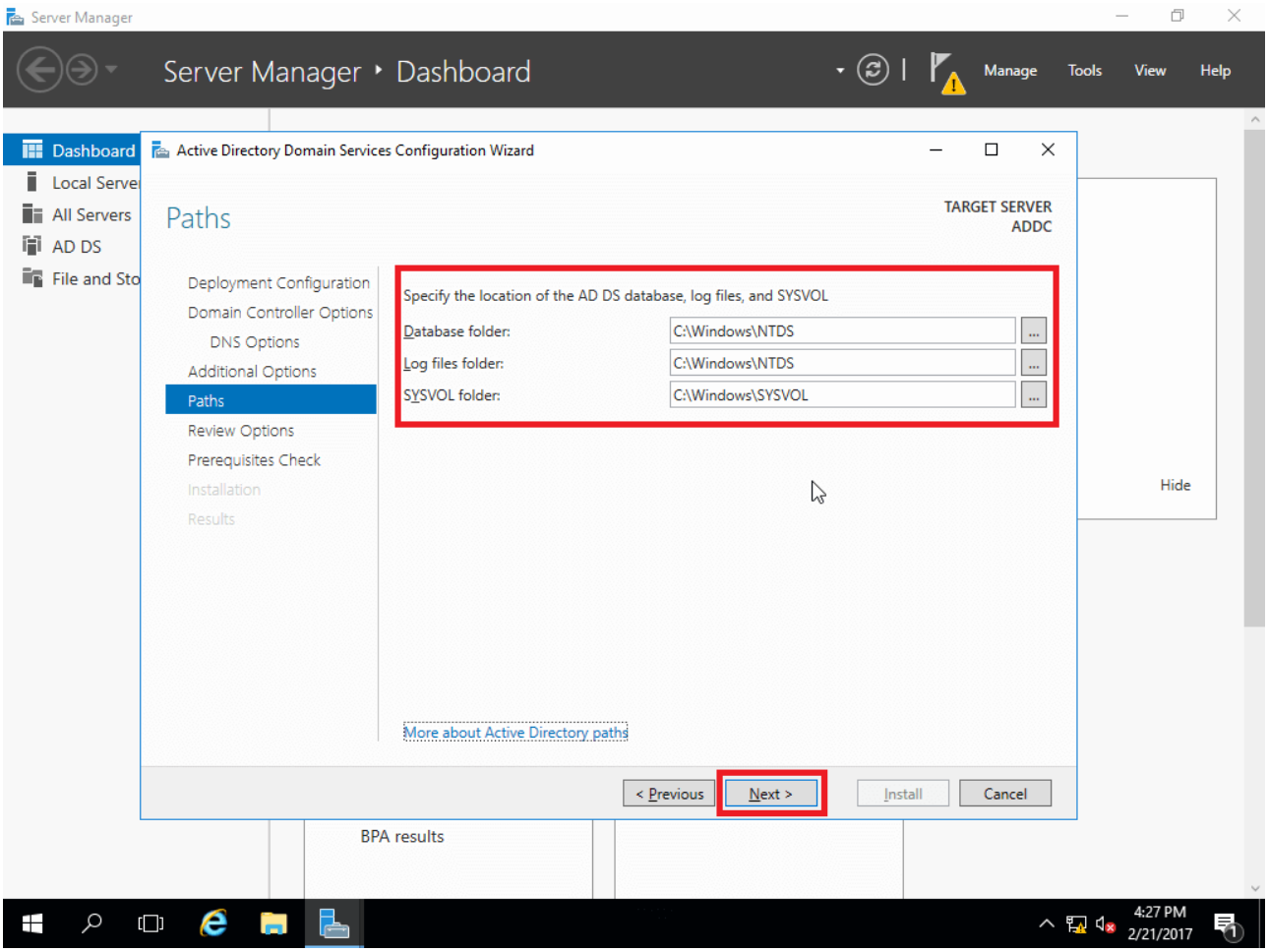
5. DNS-beállítások esetén nincs létező DNS-infrastruktúra, mivel ez az első tartományvezérlő. Tehát a figyelmeztetés figyelmen kívül hagyható. Kattintson a **Következő** gombra;



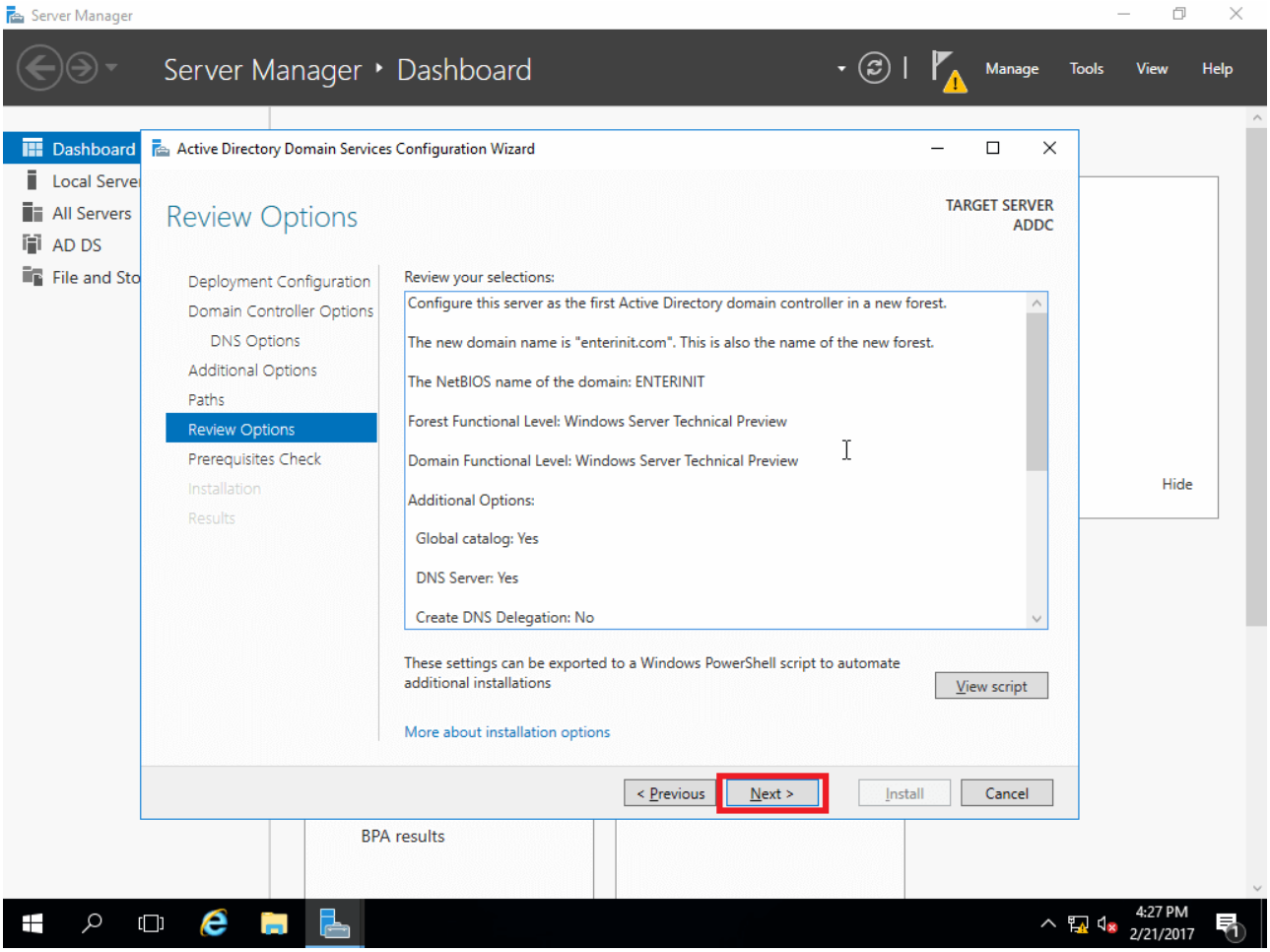
6. Adja meg a **BIOS domain nevet,** és kattintson a **Next** gombra;



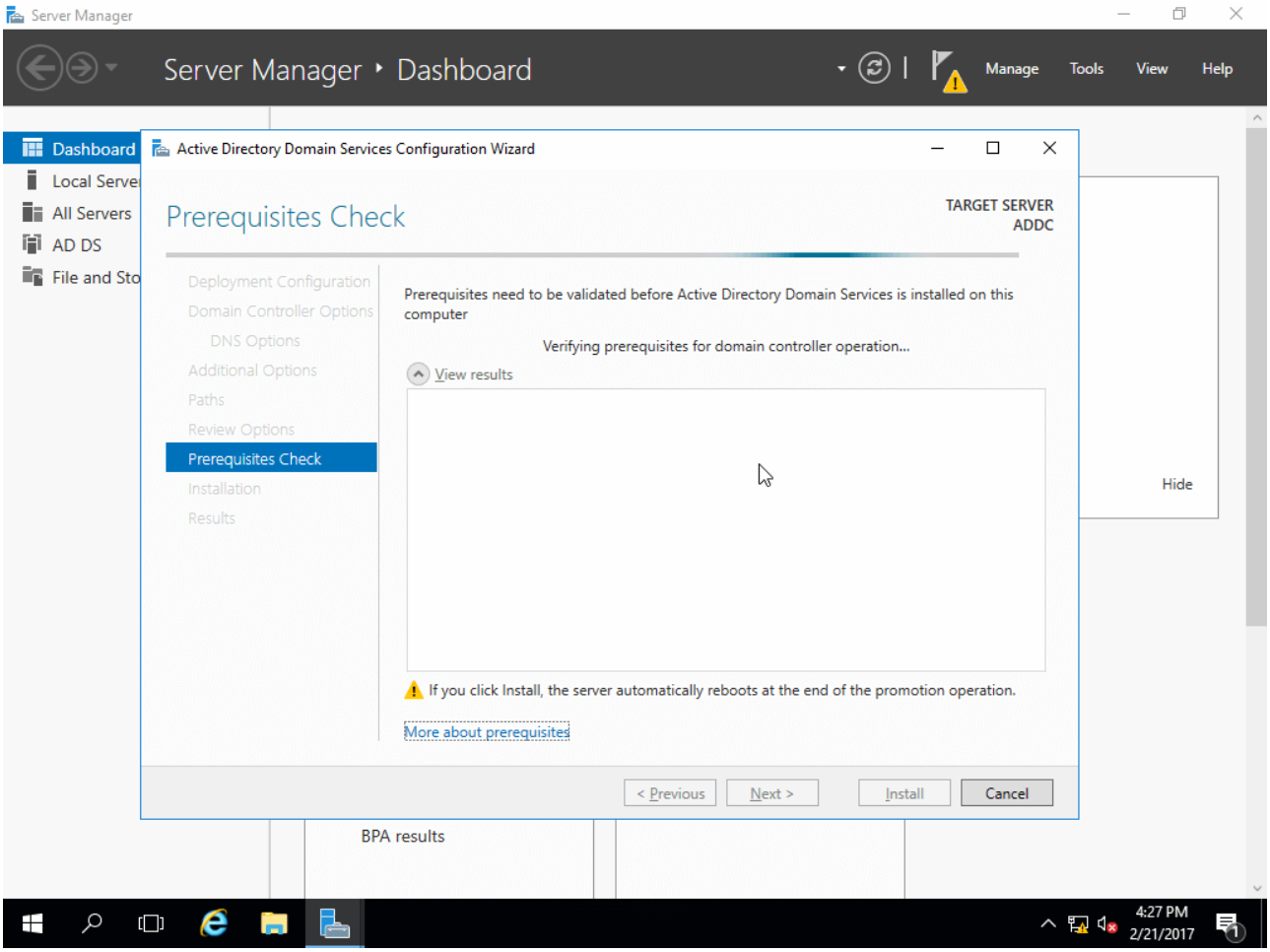
7. Ha szükséged van rá, akkor megváltoztathatja a mappák helyeit (**NEM JAVASOLT**), kattintson a **Következő** gombra;



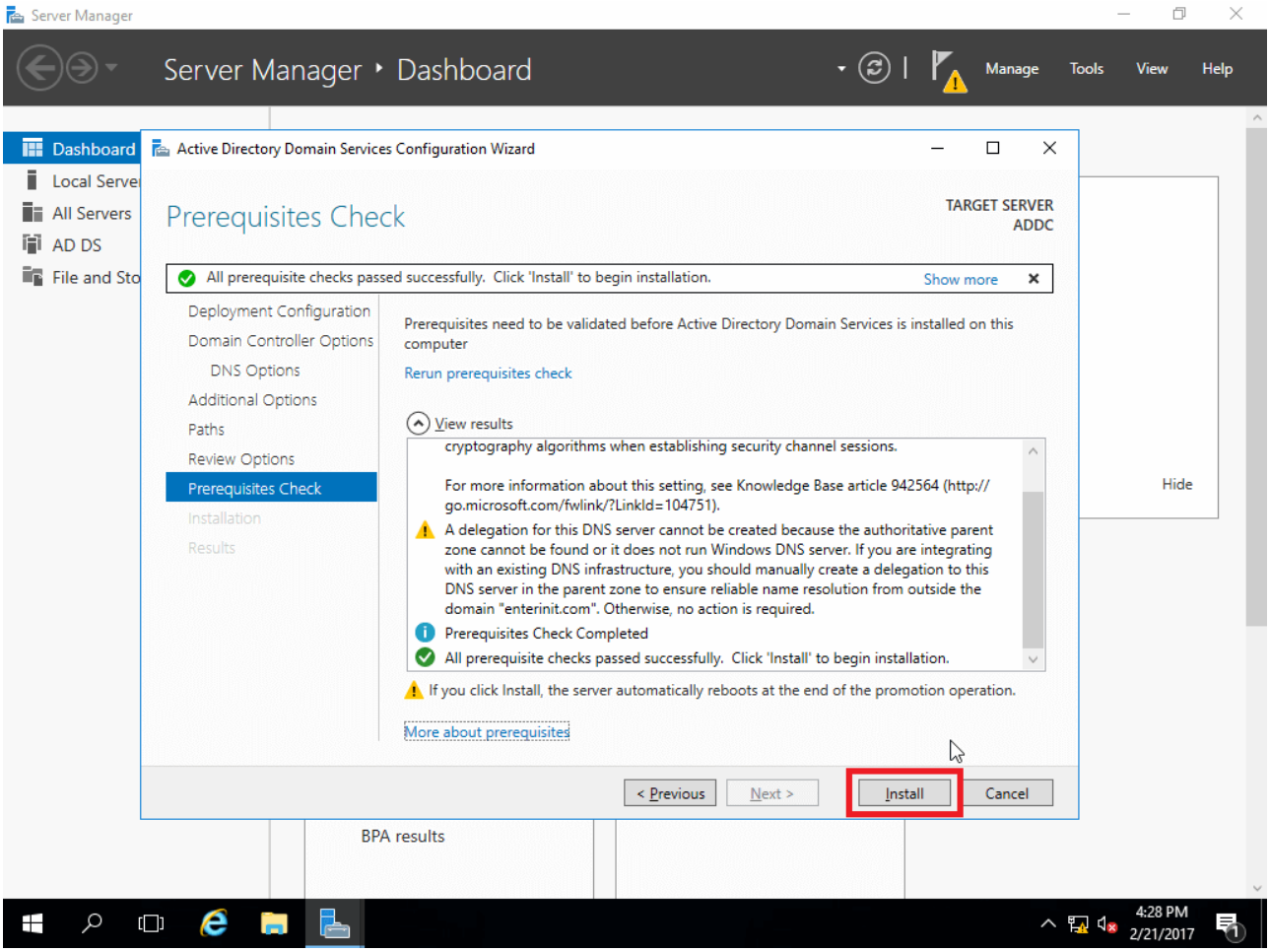
8. Felülvizsgálati lehetőségek. Kérjük, vegye figyelembe, hogy PowerShell shell parancsfájlt biztosít, ha ezt a későbbi telepítésekre kell automatizálni. Kattintson a **Script megtekintése** parancsra . Ha szükséges, másolja ezt a szkriptet későbbi felhasználásra. Zárja be a Jegyzettömb ablakot, és kattintson a **Tovább** gombra;



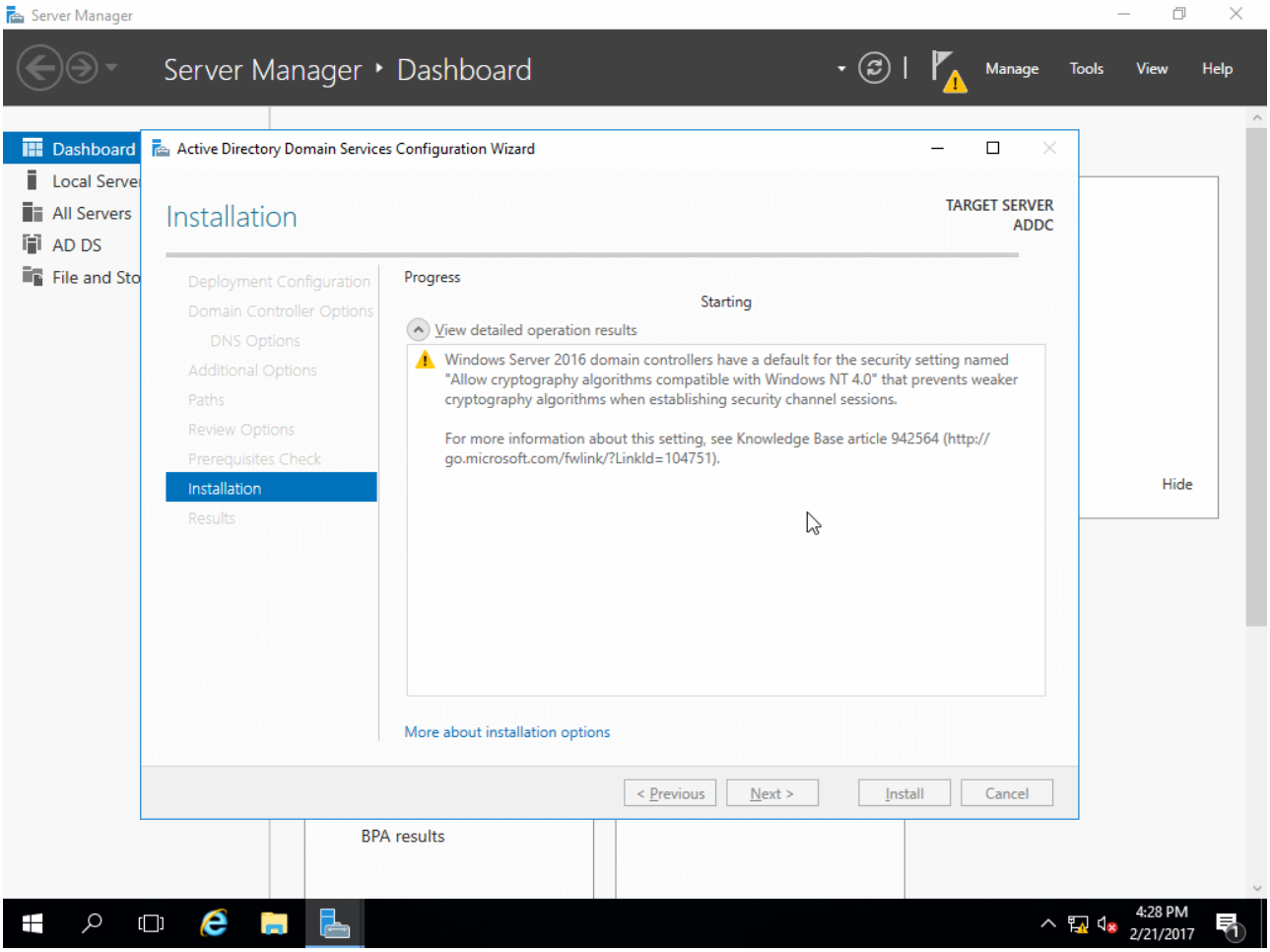
9. Rendszerellenőrzés előfeltételei;



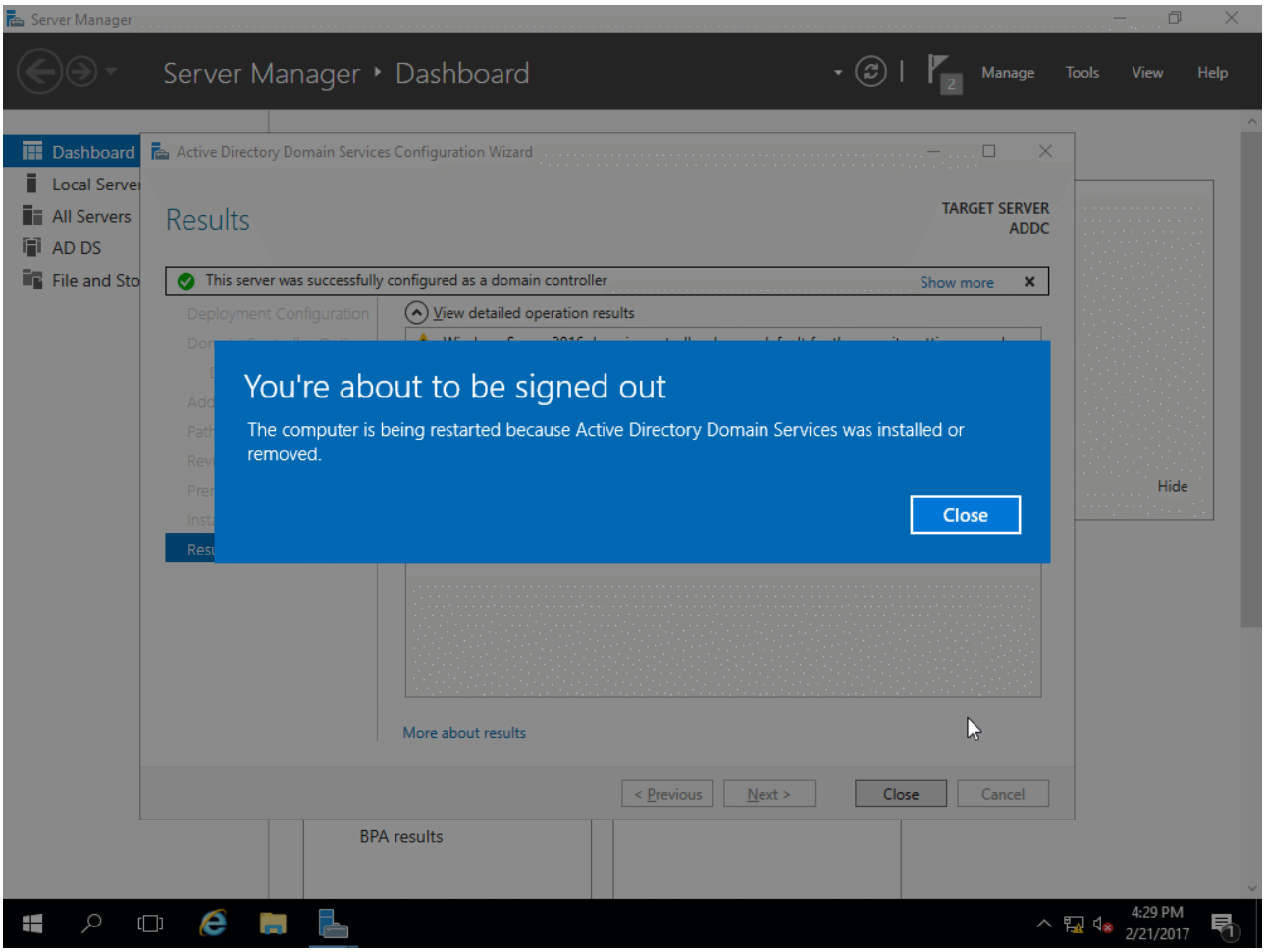
10. Ha az előfeltételek ellenőrzik az átutalásokat, kattintson az **Install** gombra;



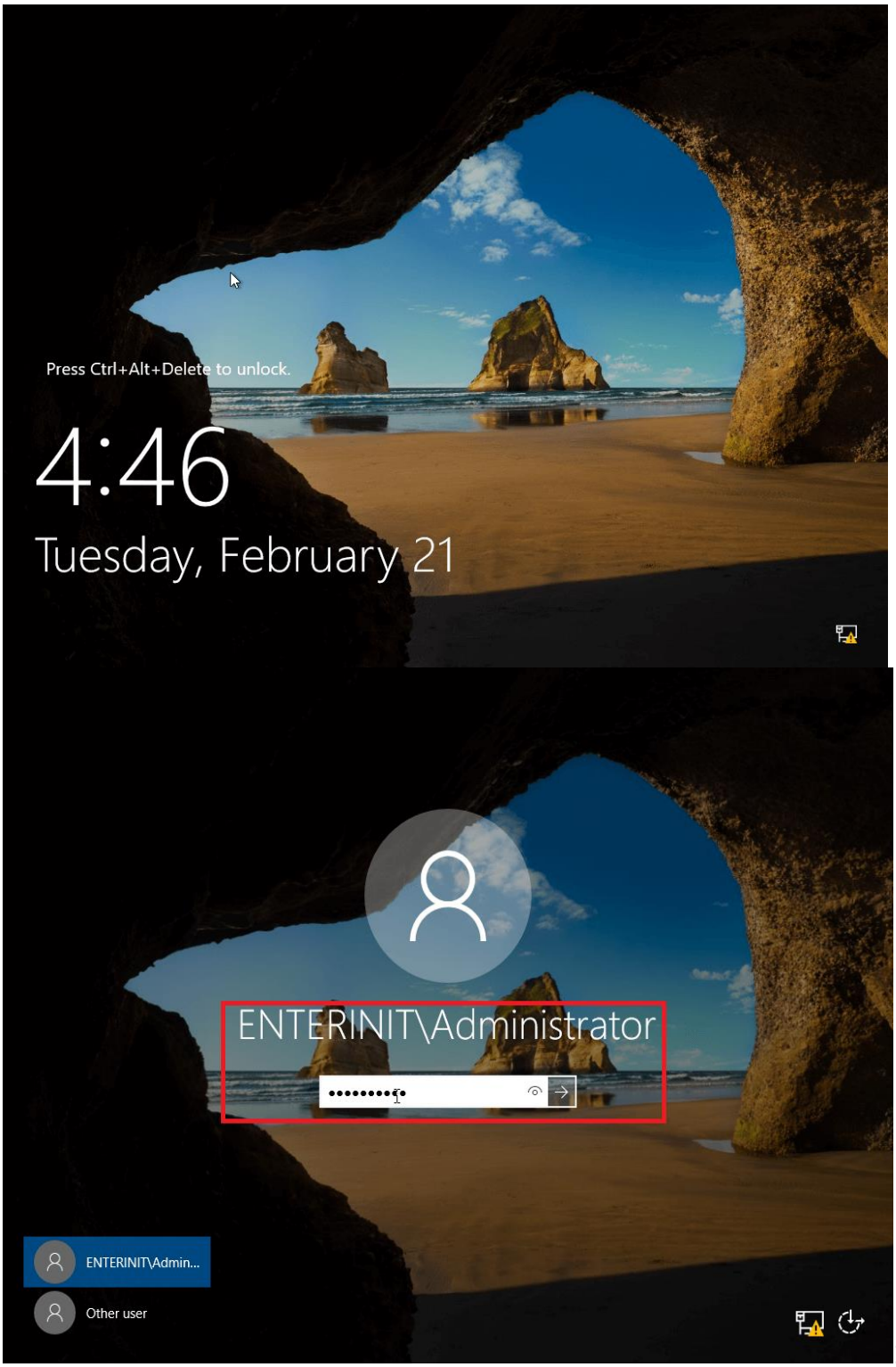
11. A telepítés megkezdődik;



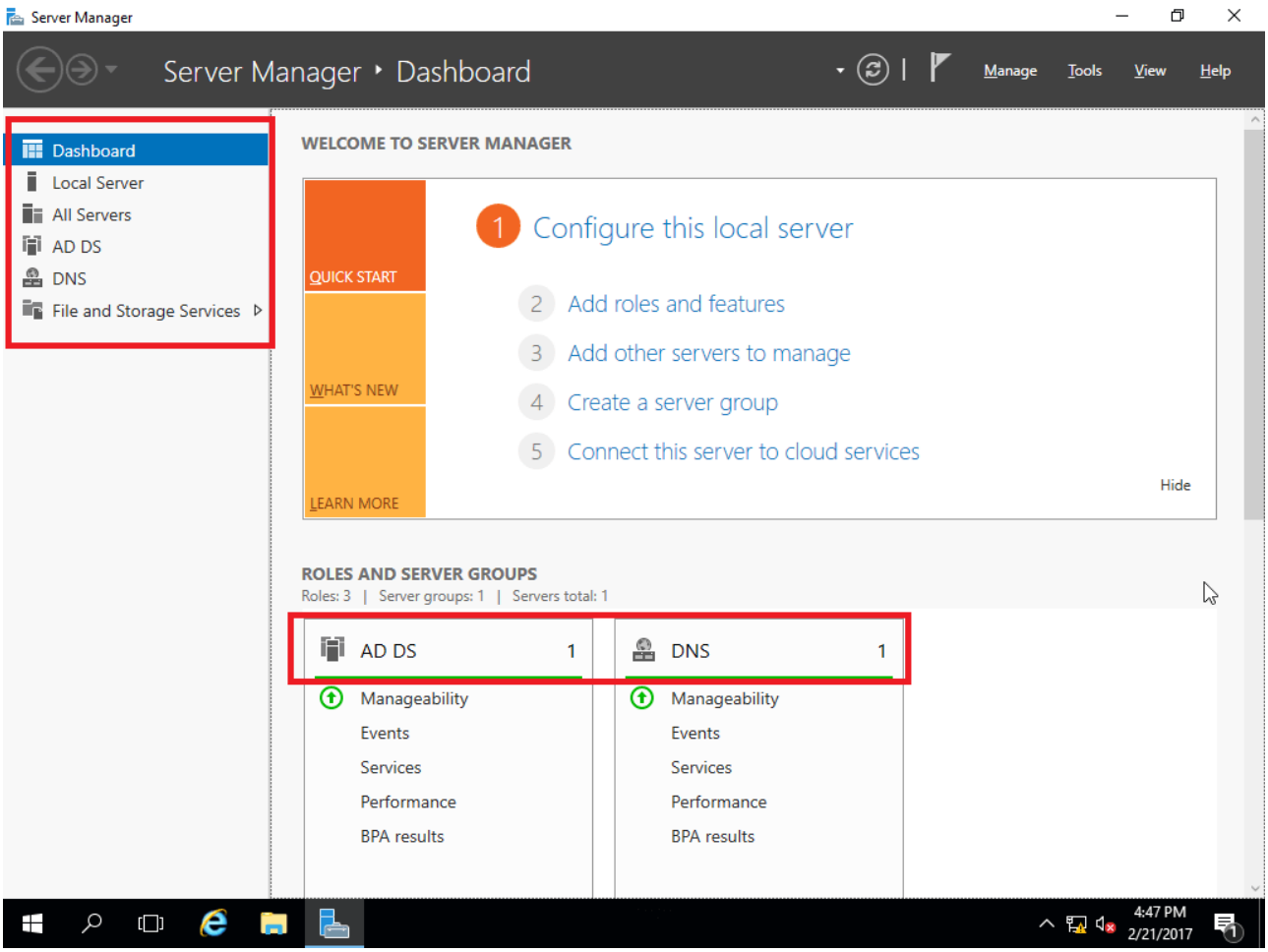
12. A teljes rendszer újraindítása után;



13. Az újraindítás után bejelentkezhet a PC-re, mint **Domain Admin** ;

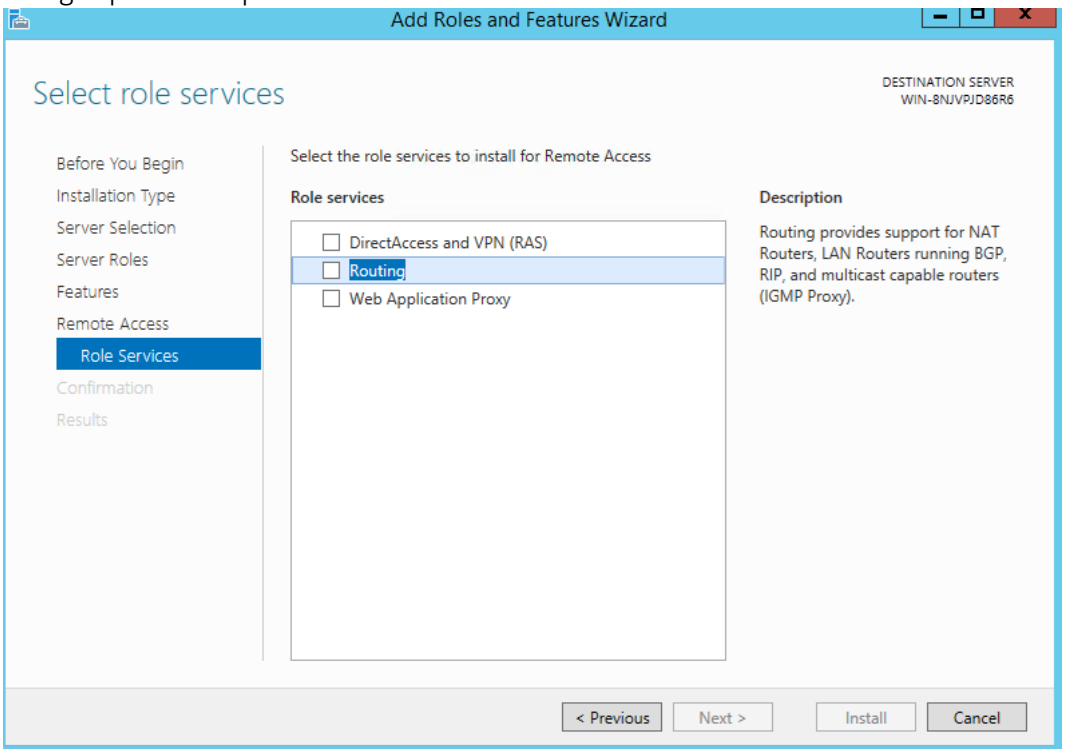


14. Ez a Windows Server 2016 Active Directory telepítésének eredménye;


Server Manager
Server Manager • Dashboard
Manage
Tools
View
Help
Dashboard
Local Server
All Servers
AD DS
DNS
File and Storage Services
WELCOME TO SERVER MANAGER
QUICK START
WHAT'S NEW
LEARN MORE
1 Configure this local server
2 Add roles and features
3 Add other servers to manage
4 Create a server group
5 Connect this server to cloud services
Hide
ROLES AND SERVER GROUPS
Roles: 3 | Server groups: 1 | Servers total: 1
AD DS 1
Manageability
Events
Services
Performance
BPA results
DNS 1
Manageability
Events
Services
Performance
BPA results
4:47 PM
2/21/2017

A Windows Server 2016 R2 routing / NAT funkcióinak beállítása

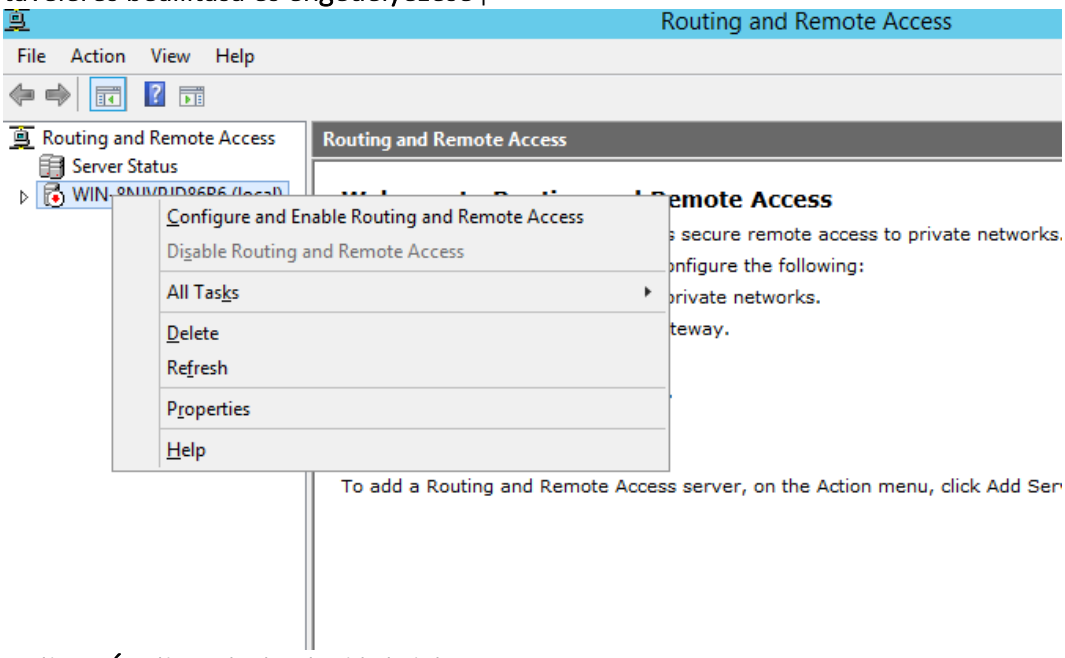
1. A kiszolgálókezelő elindításához kattintson a **Manage> Add Roles and Features** elemre .
2. Az Előkészületek lapon kattintson a **Tovább** gombra .
3. Válassza a **Szerepkör alapú vagy funkcióalapú telepítést,** majd kattintson a **Tovább** gombra .
4. A Szerver kiválasztása mezőben válassza ki azt a kiszolgálót, amelyre telepíteni kívánja a szolgáltatást, majd kattintson a **Tovább** gombra .
5. A kiszolgálói szerepkörök listájában válassza a Távoli hozzáférés lehetőséget, majd kattintson a **Tovább**gombra .
6. A Szolgáltatások oldalon kattintson a **Tovább** gombra .
7. Kattintson a **Tovább** a Távoli hozzáférés lehetőségre.
8. A Szerepkörnyezet szolgáltatásairól kattintson az **Útválasztás** jelölőnégyzetre. Kattintson a **Funkciók hozzáadása** lehetőségre az előugró párbeszédpanelen.



1. ábra: Szerepkörök és szolgáltatások varázsló hozzáadása

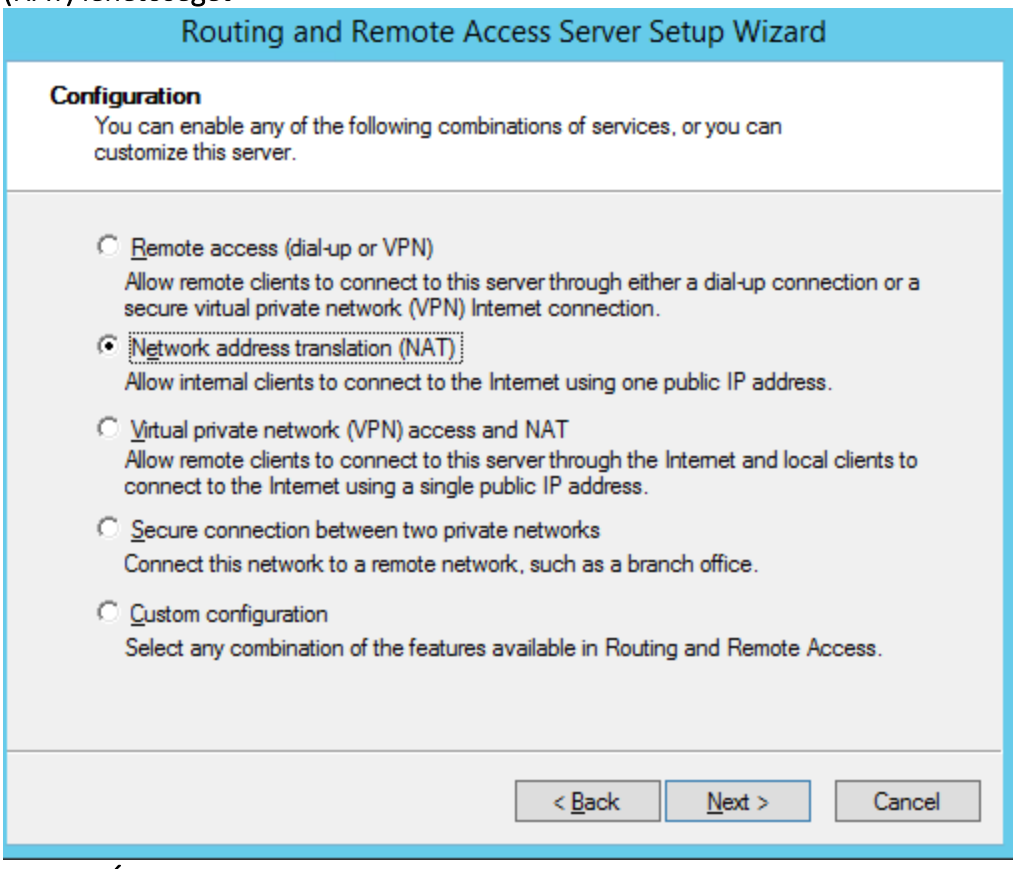
9. Kattintson a **Tovább** gombra .
10. Kattintson a **Tovább** gombra a Webszerver szerepkör (IIS) oldalon.
11. Ezen az Illetmény szolgáltatások oldalon hagyja el az alapértelmezett értékeket, majd kattintson a **Tovább**gombra .
12. Végül kattintson a Confirmation (**Telepítés)** gombra .
13. A funkció hozzá lett adva a Windows rendszerhez. Ez a folyamat nem szükséges újbóli indításhoz, ezért most továbblépünk a konfiguráció következő részére.
14. A **Felügyeleti eszközökben** keresse meg az **Útválasztás és távelérés lehetőséget** . Nyissa meg a konzolt, és piros lefelé mutató nyíl jelenik meg a kiszolgáló nevén.

15. Kattintson a jobb gombbal a kiszolgáló nevére, majd kattintson az Útválasztás és távelérés beállítása és engedélyezése parancsra .



2. ábra: Útválasztás és távelérés képernyő

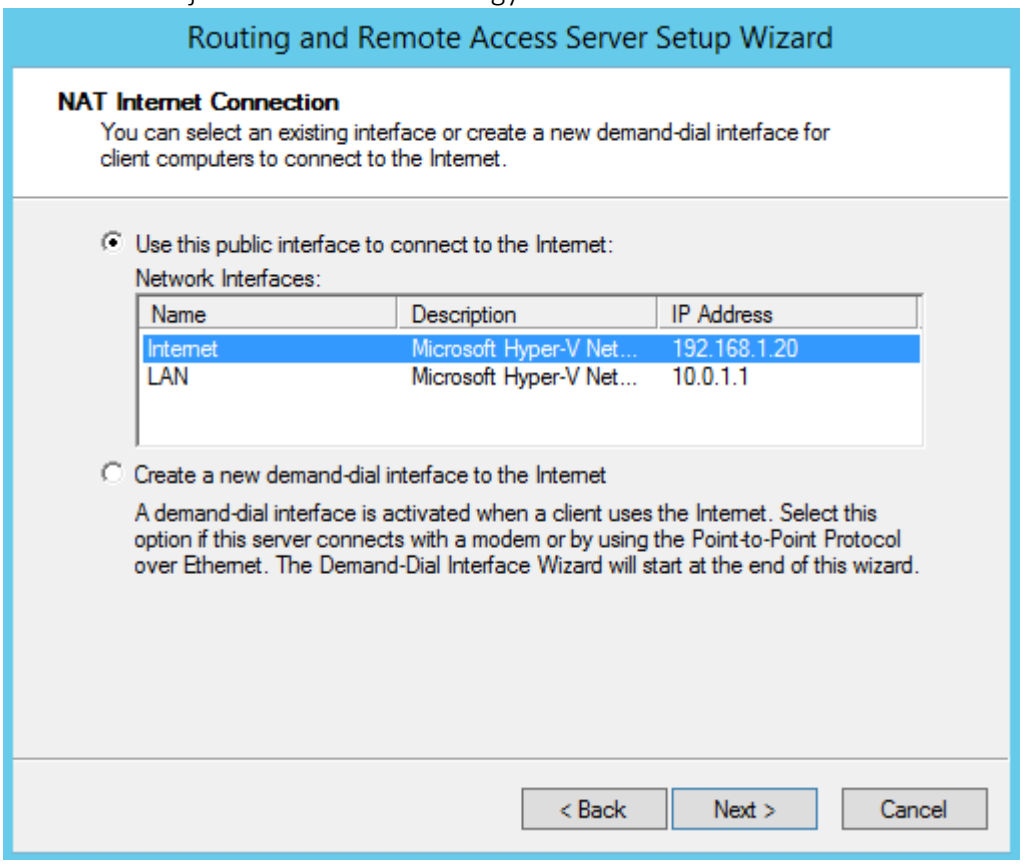
16. Megjelenik az Útválasztó és távelérési kiszolgáló telepítővarázslója.
17. Kattintson a Tovább gombra .
18. A Configuration (Konfiguráció) képernyőn válassza a Network Address Translation (NAT) lehetőséget .



3. ábra: Útválasztó és távelérési kiszolgáló telepítővarázslója

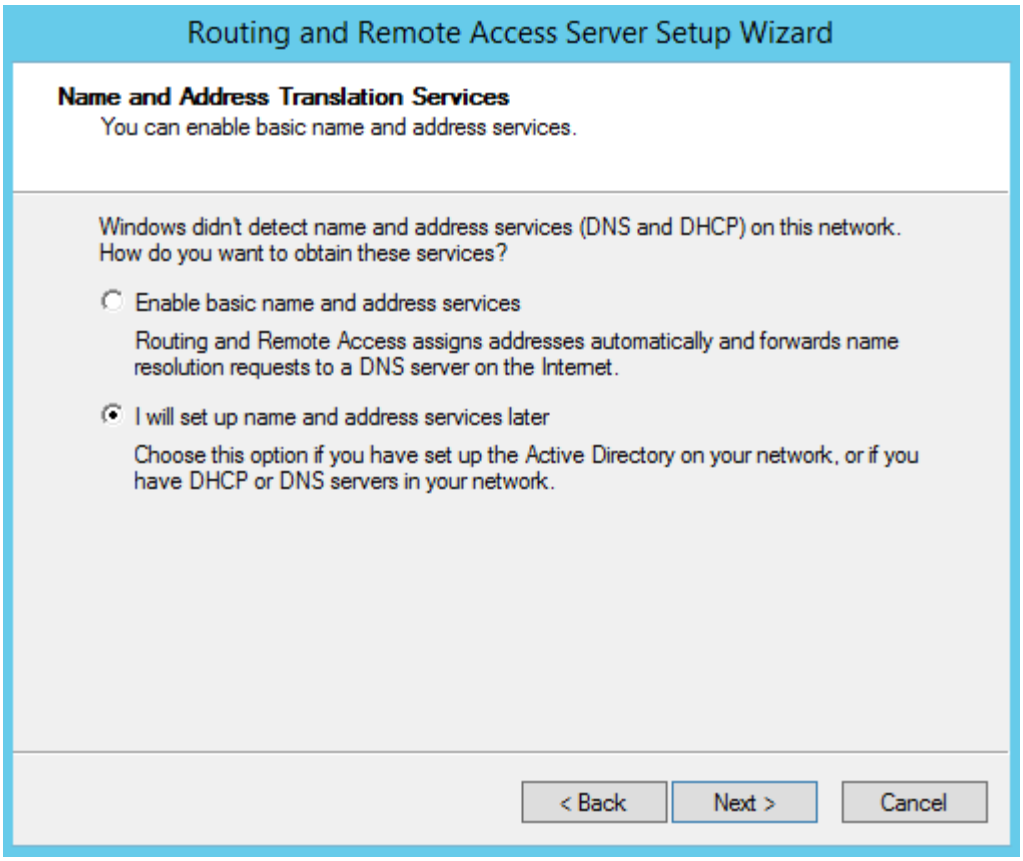
Megjegyzés : Szükség esetén kiválaszthatja a VPN & NAT beállítást, ha VPN funkciókat szeretne beállítani a rendszeren is.

19. Kattintson a **Tovább** gombra .
20. Itt kiválaszthatja a külső hálózathoz vagy az internethez csatlakozó hálózati interfészt.



4. ábra: Útválasztás és távelérés hálózati csatoló kiválasztása

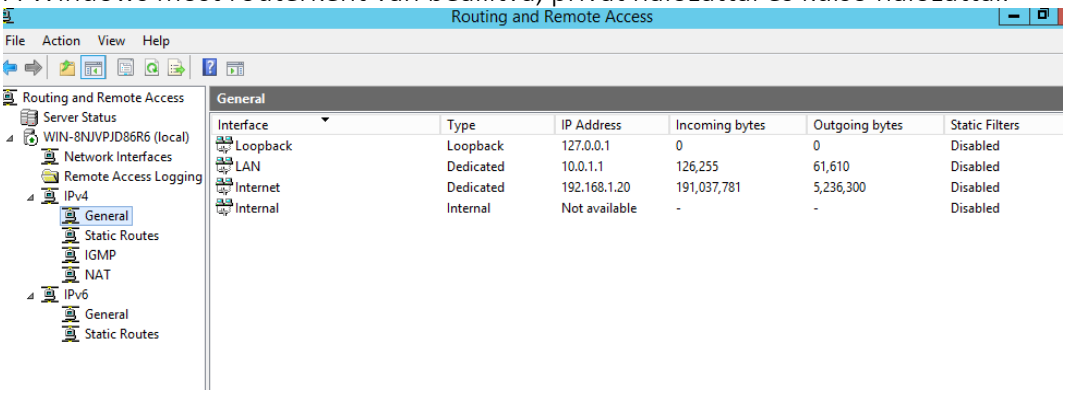
21. Kattintson a **Tovább** gombra .
22. Ezután el kell választania, hogy az RRAS biztosítsa a DHCP és a DNS átirányítását, vagy más módon válassza ki, hogy később, ha szükséges, beállítja a DHCP és DNS hálózatát.
23. Konfigurációnkban a második opciót választjuk, mivel a legtöbb esetben a DNS-t és a DHCP-t külön beállítjuk az RRAS-től.



5. ábra: Útválasztás és távelérési NAT szolgáltatások

24. Kattintson a **Tovább** gombra .
25. Kattintson a **Befejezés** gombra .

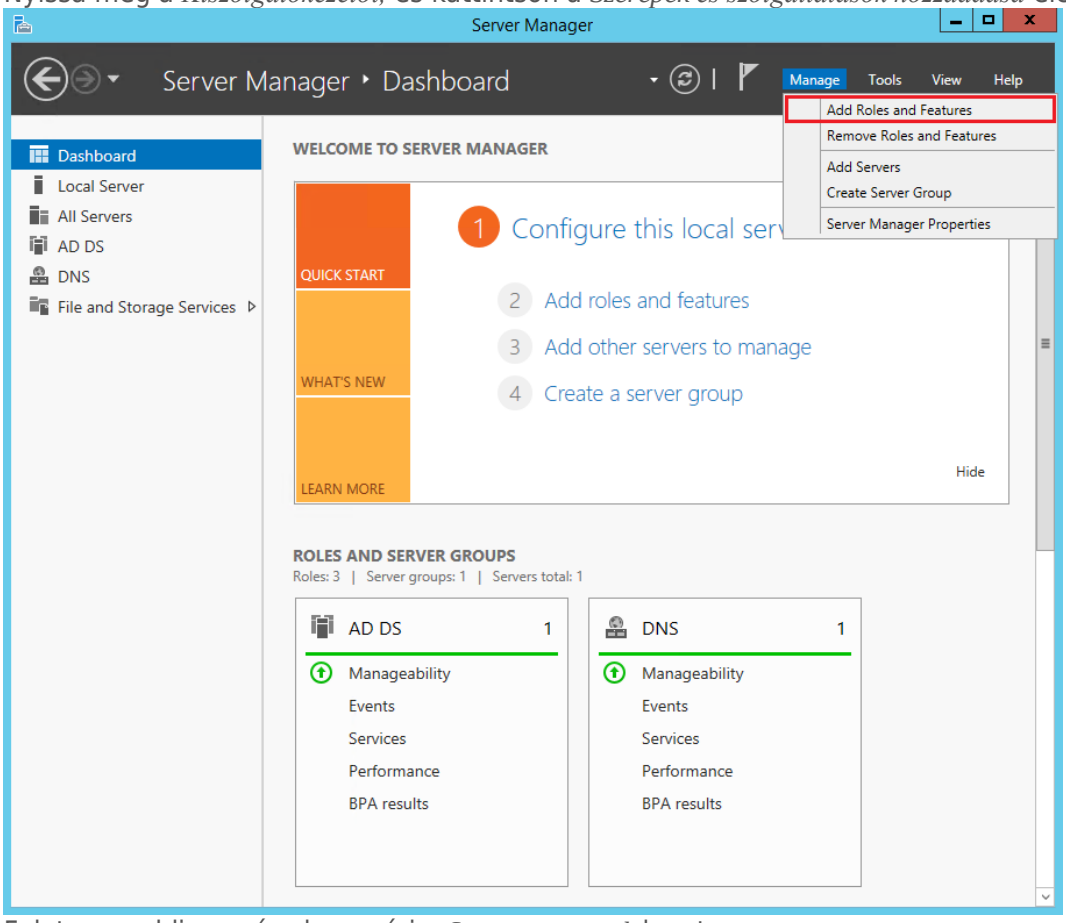
A Windows most routerként van beállítva, privát hálózattal és külső hálózattal.



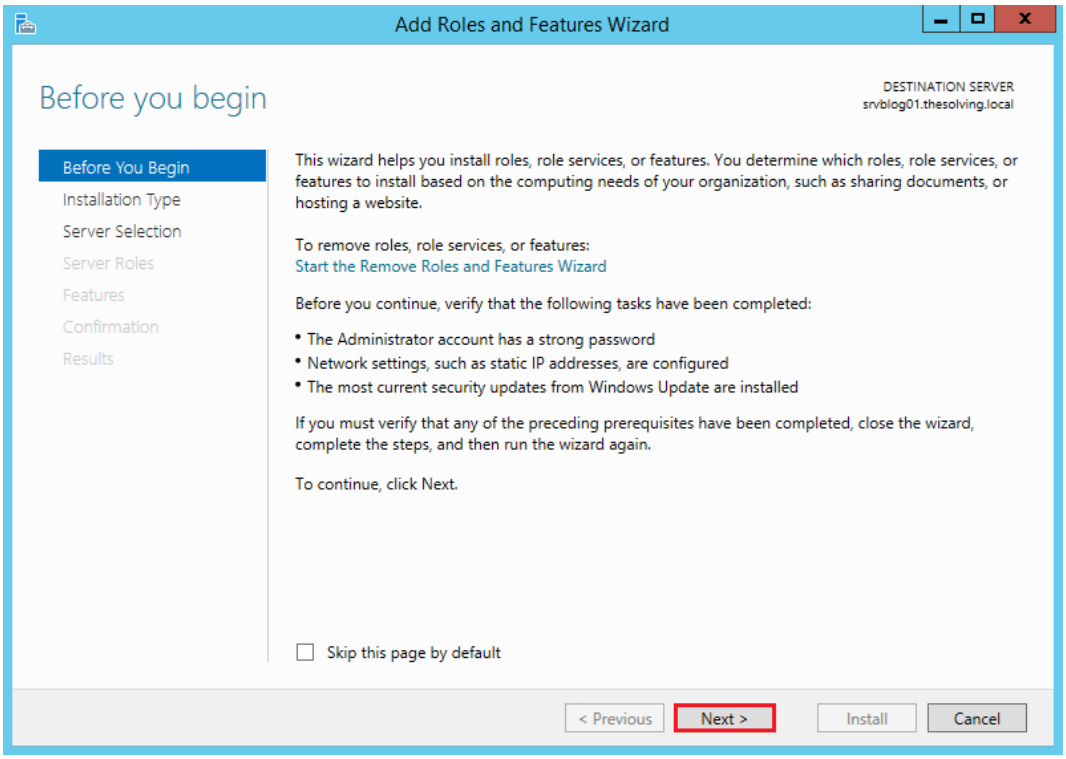
6. ábra: A telepítés magán és külső hálózatokkal

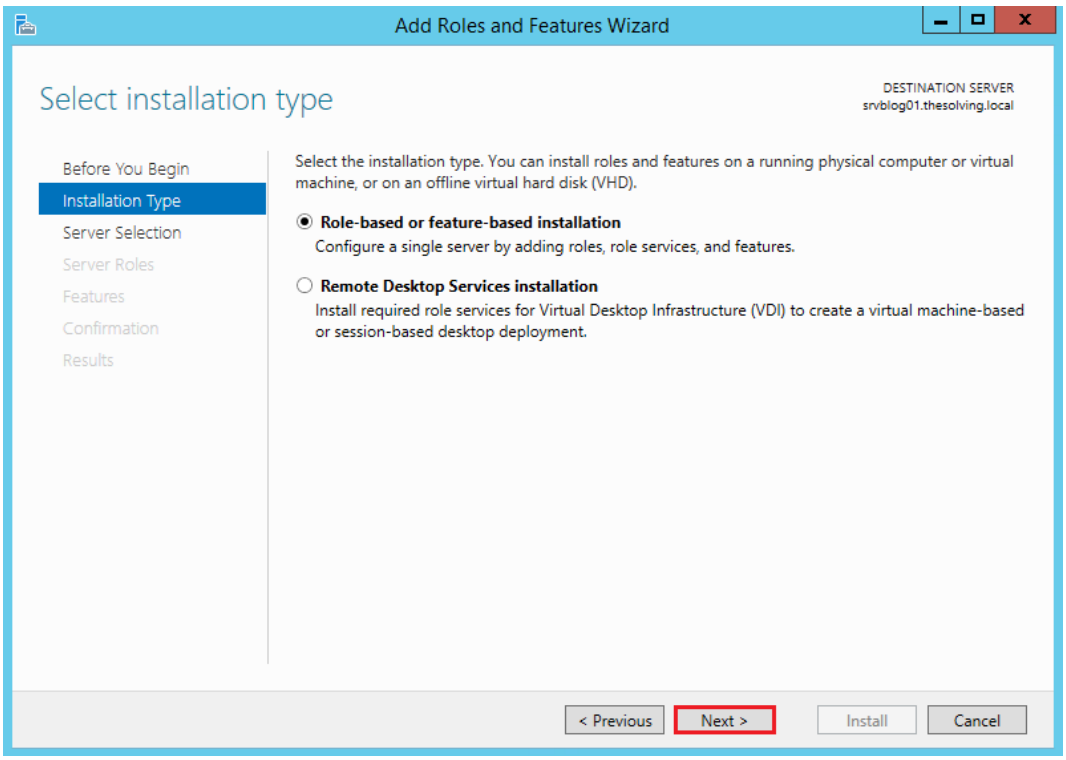
Első lépés: Telepítse a webszerver (IIS) szerepét

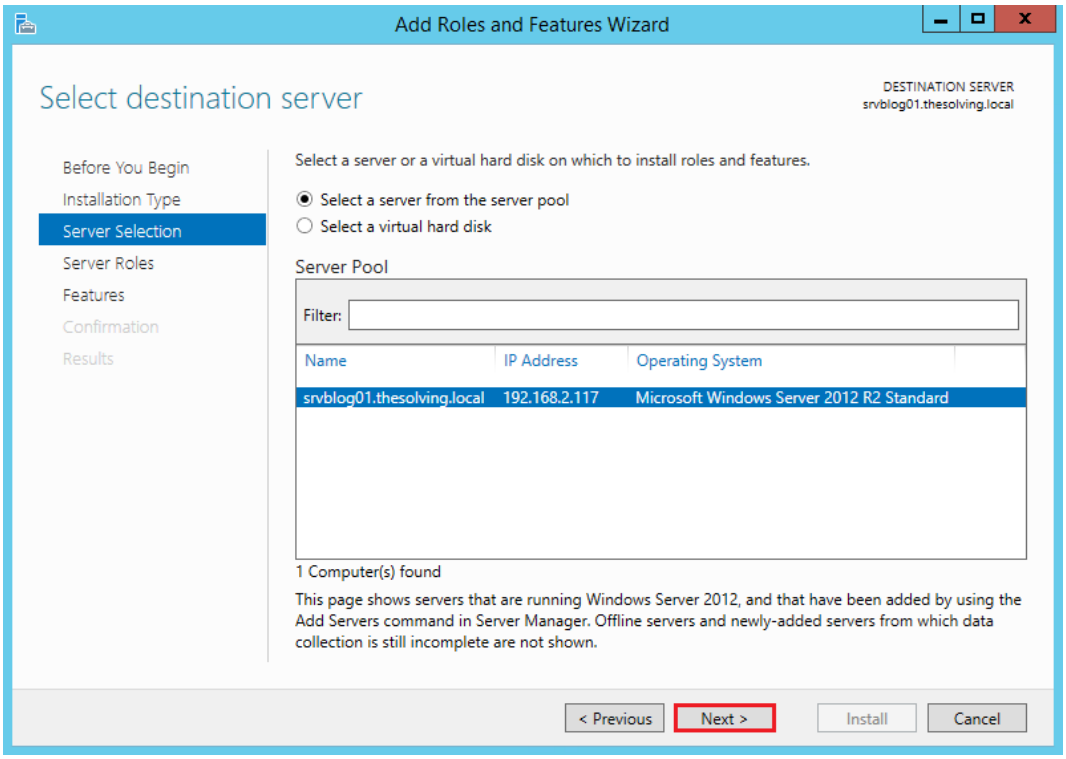
Nyissa meg a *Kiszolgálókezelőt,* és kattintson a *Szerepek és szolgáltatások hozzáadása* elemre *:*



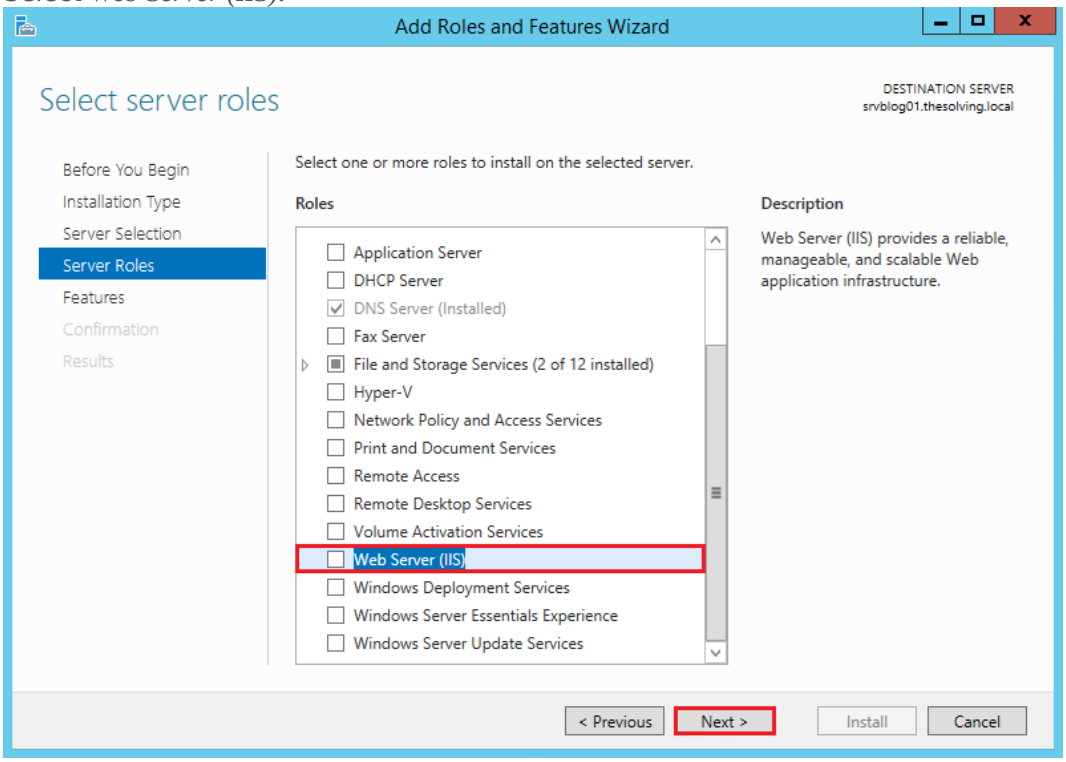
Folytassa addig, amíg el nem éri a *Szerver szerepek* lapot:


Add Roles and Features Wizard
Before you begin
DESTINATION SERVER
srvblog01.thesolving.local
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
This wizard helps you install roles, role services, or features. You determine which roles, role services, or features to install based on the computing needs of your organization, such as sharing documents, or hosting a website.
To remove roles, role services, or features:
Start the Remove Roles and Features Wizard
Before you continue, verify that the following tasks have been completed:
• The Administrator account has a strong password
• Network settings, such as static IP addresses, are configured
• The most current security updates from Windows Update are installed
If you must verify that any of the preceding prerequisites have been completed, close the wizard, complete the steps, and then run the wizard again.
To continue, click Next.
Skip this page by default
< Previous
Next >
Install
Cancel

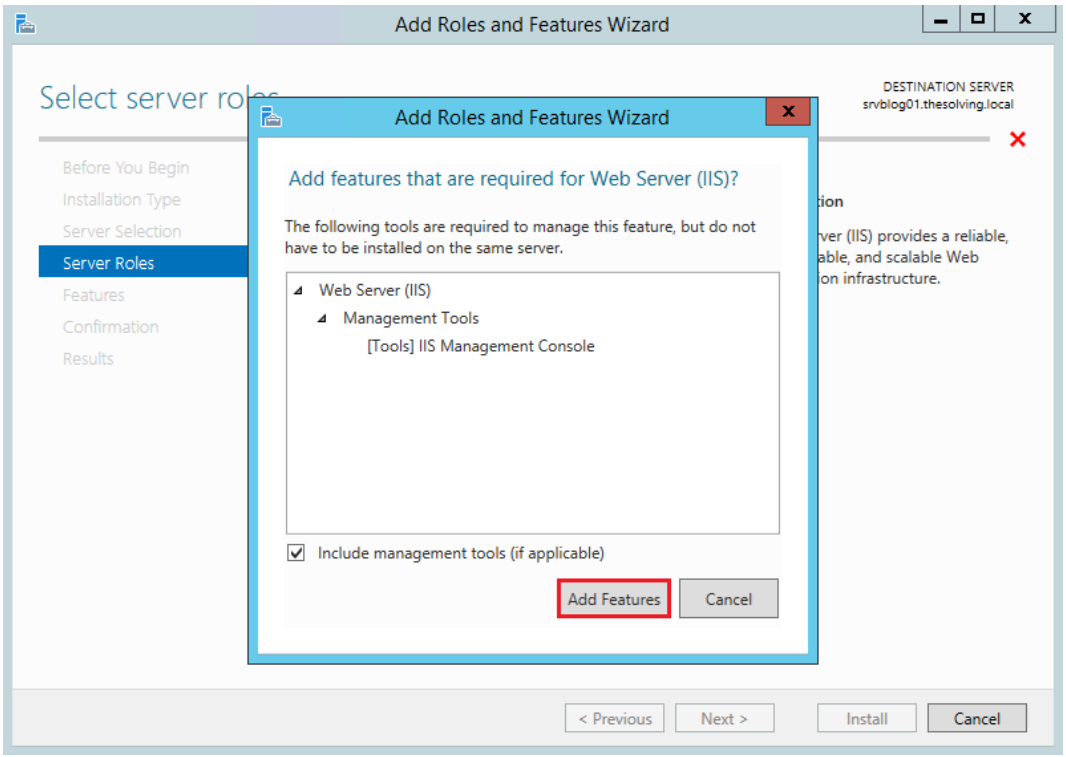

Add Roles and Features Wizard
Select installation type
DESTINATION SERVER
srvblog01.thesolving.local
Before You Begin
Installation Type
Server Selection
Server Roles
Features
Confirmation
Results
Select the installation type. You can install roles and features on a running physical computer or virtual machine, or on an offline virtual hard disk (VHD).
Role-based or feature-based installation
Configure a single server by adding roles, role services, and features.
Remote Desktop Services installation
Install required role services for Virtual Desktop Infrastructure (VDI) to create a virtual machine-based or session-based desktop deployment.
< Previous
Next >
Install
Cancel



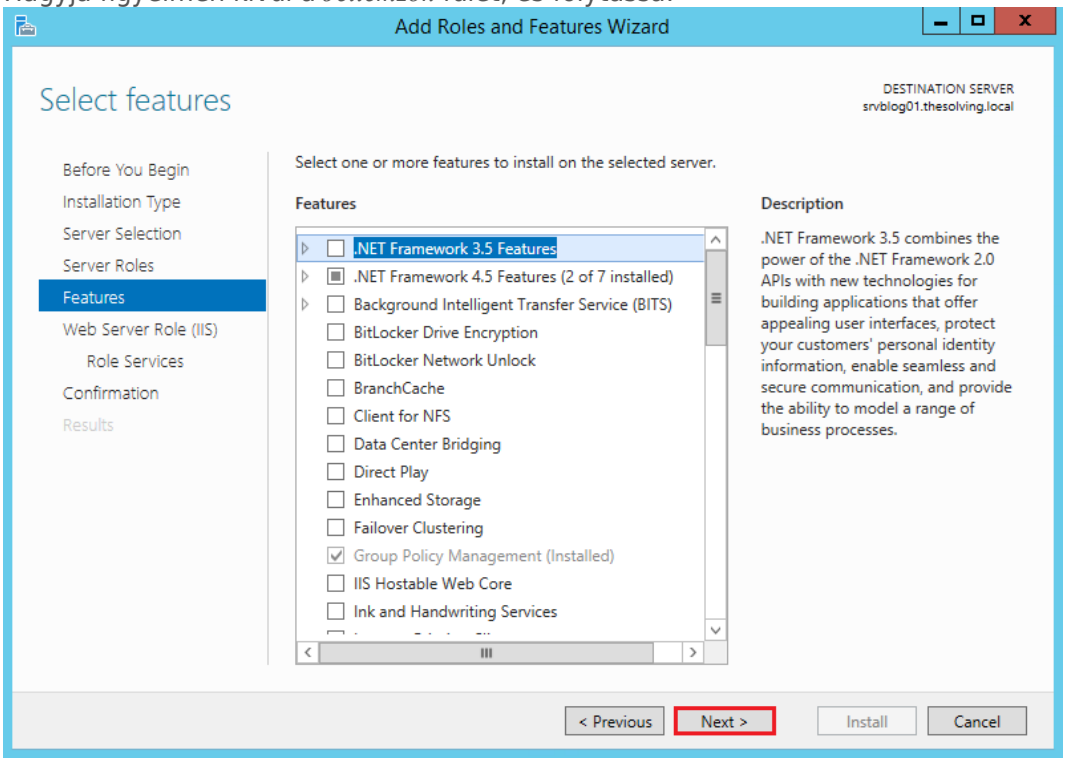
Select *Web Server (IIS):*



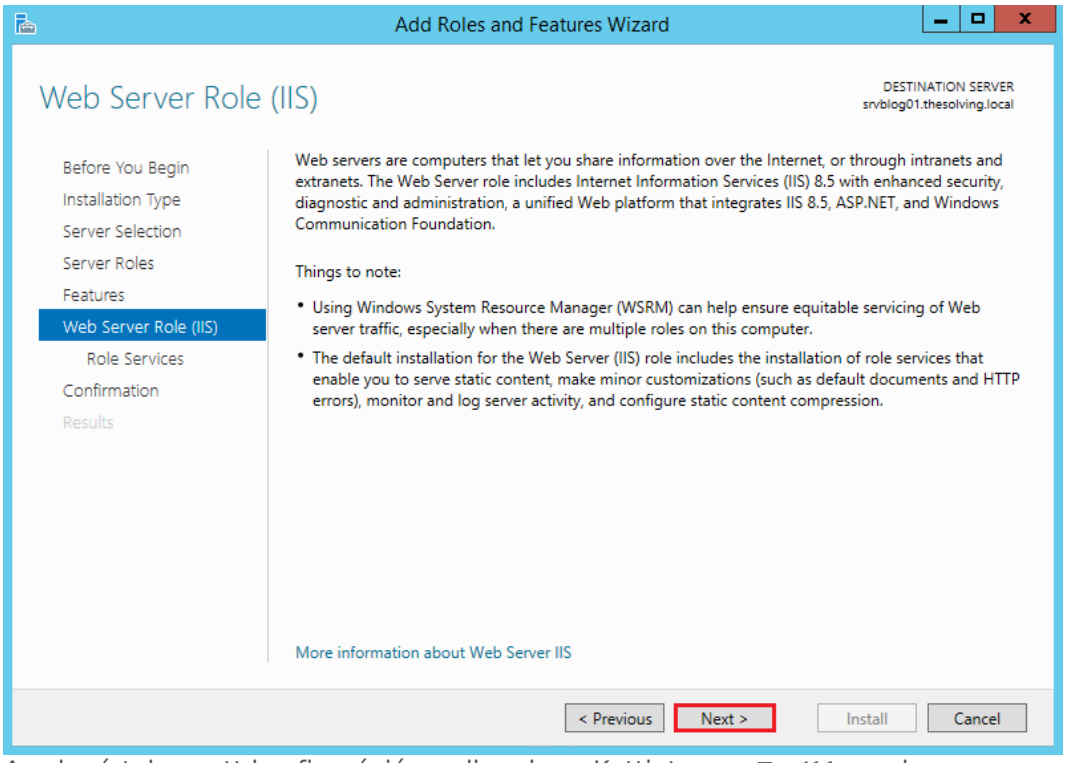
Kattintson a *Szolgáltatások hozzáadása* elemre *:*



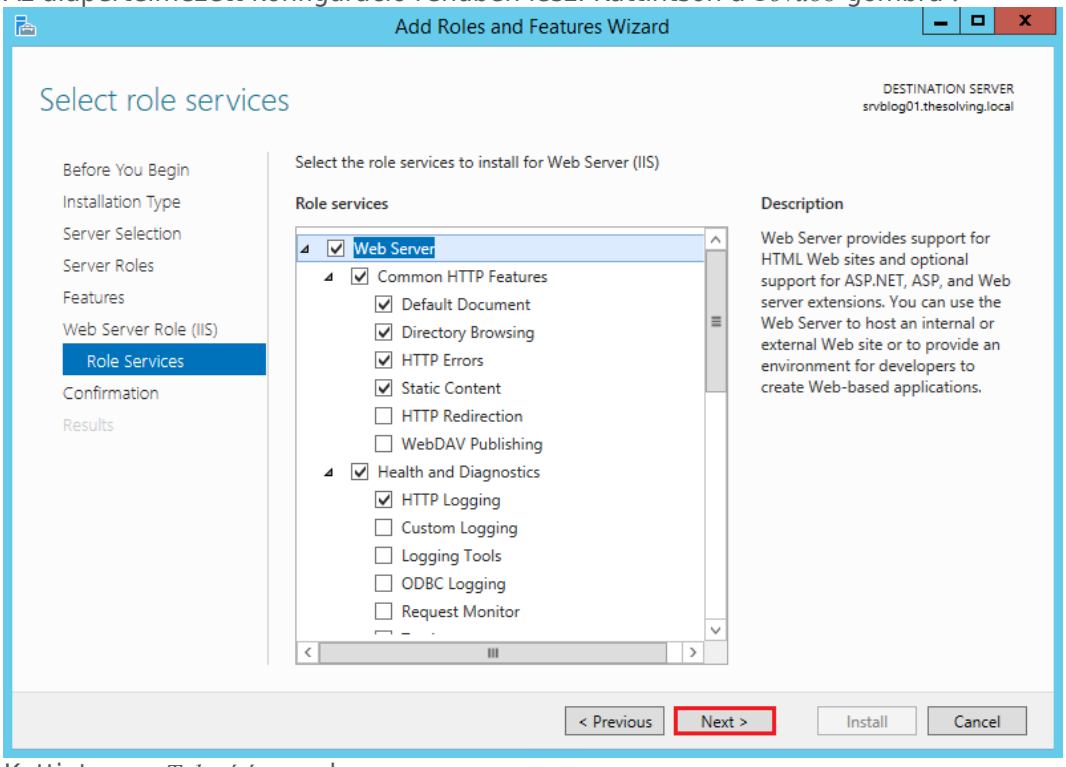
Hagyja figyelmen kívül a *Jellemzők* fület, és folytassa:



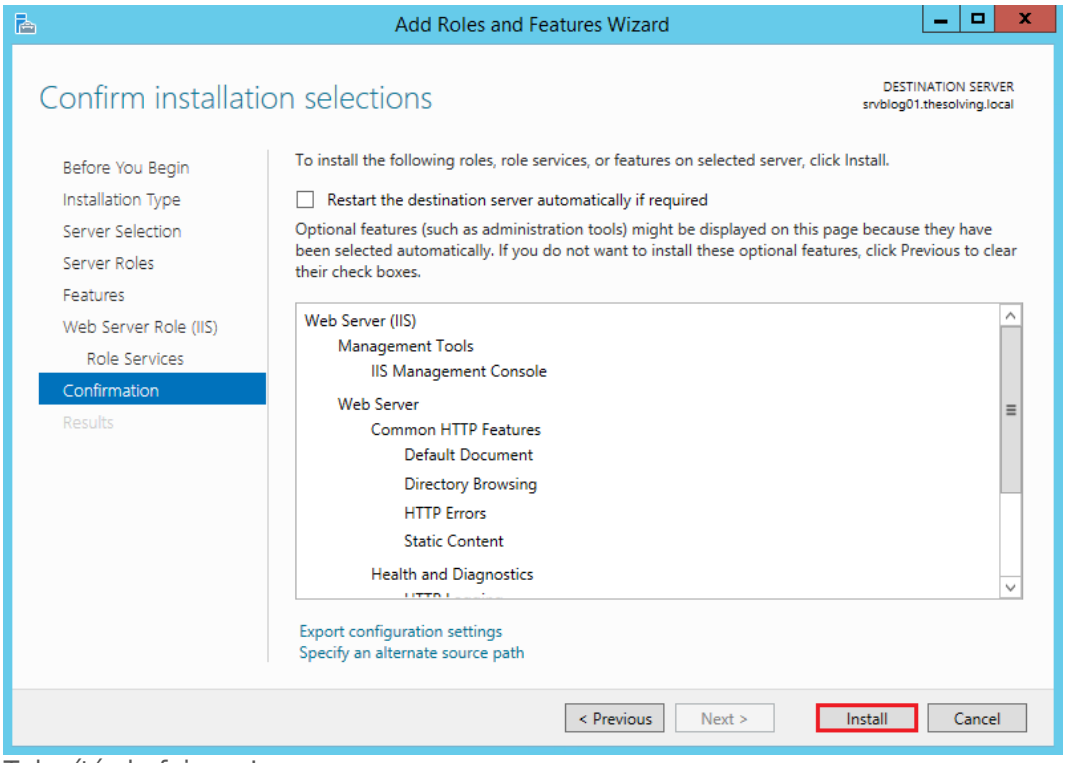
Kattintson a *Tovább* gombra :



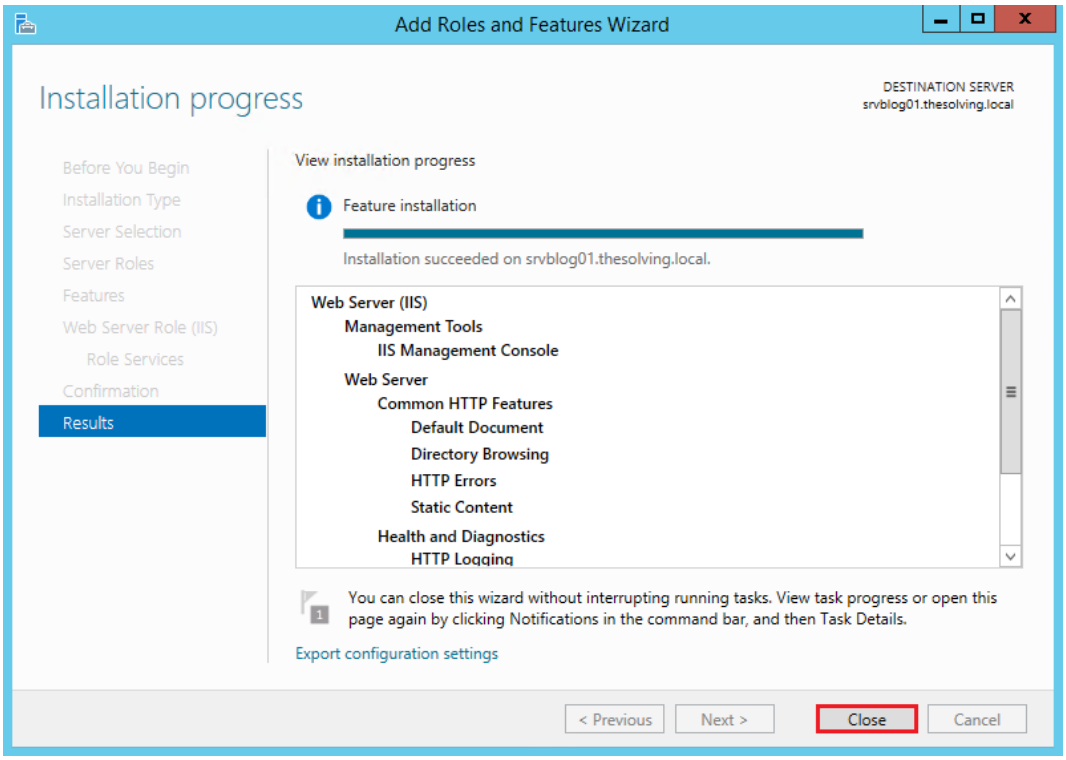
Az alapértelmezett konfiguráció rendben lesz. Kattintson a *Tovább* gombra *:*



Kattintson a *Telepítés* gombra *:*

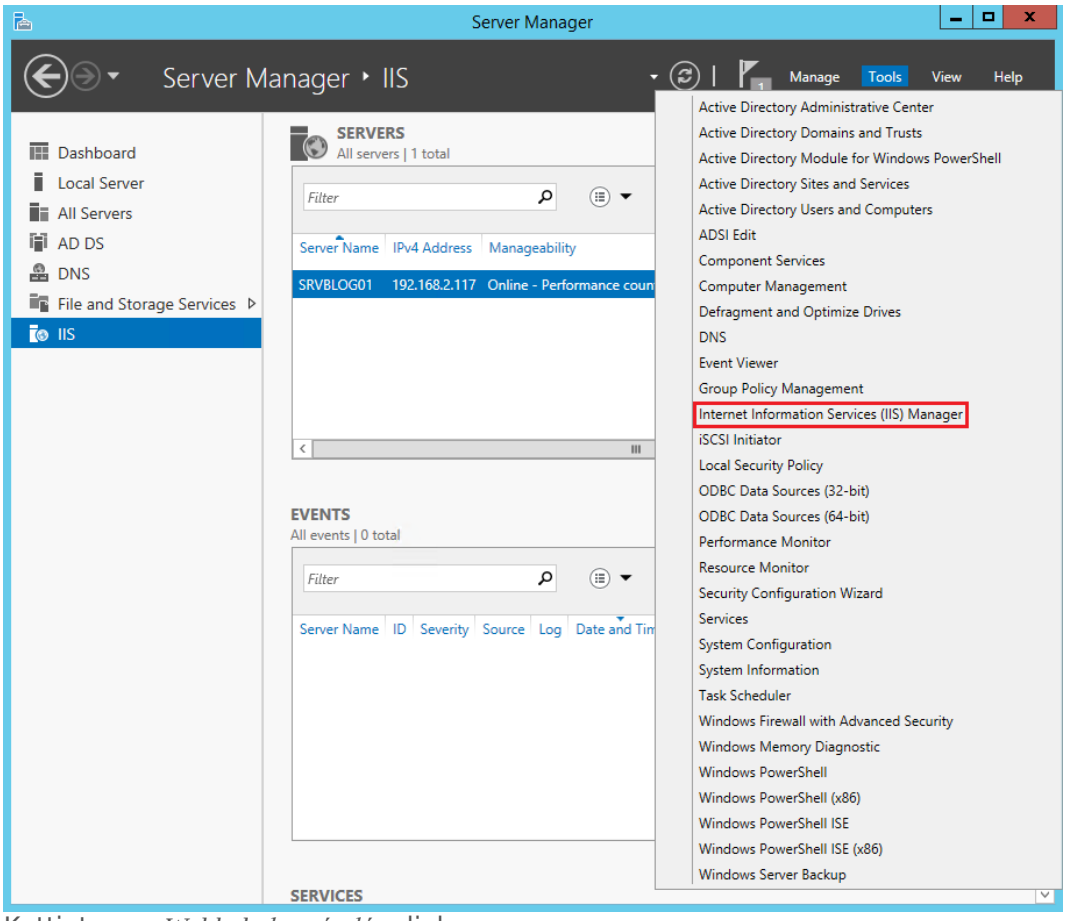


Telepítés befejezve!

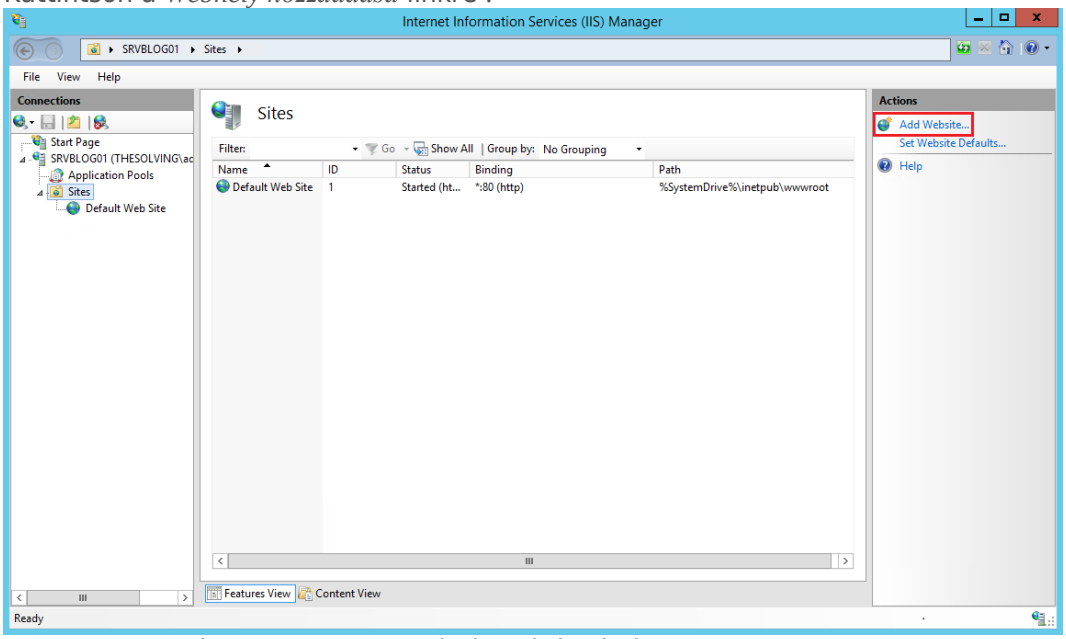


Második lépés: Az IIS konfigurálása

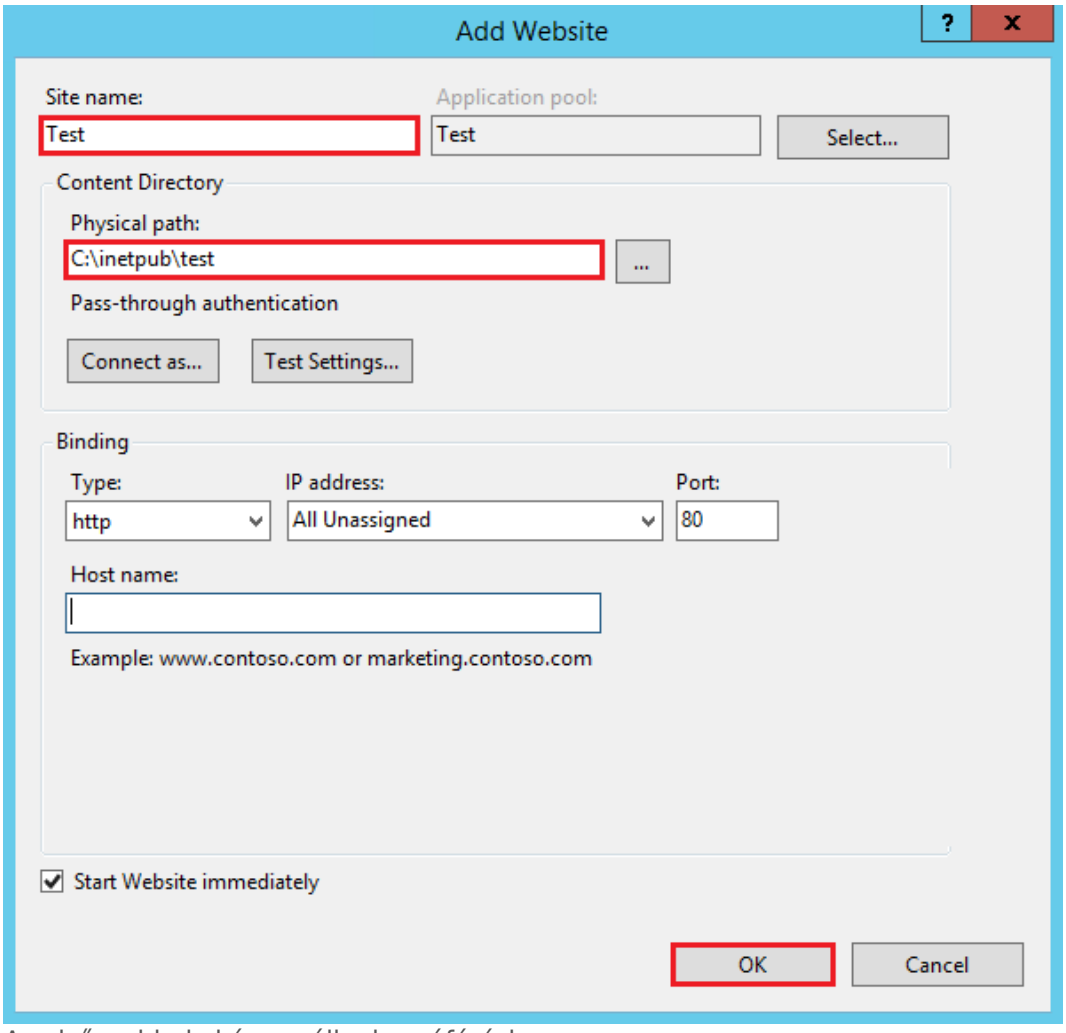
Menjen vissza a *Kiszolgálókezelőhöz* . Válassza *az Internet Information Services (IIS) kezelője* a *Tools* menü:



Kattintson a *Webhely hozzáadása* linkre *:*



Adja meg legalább a webhely nevét és elérési útját. Kattintson az *OK* gombra *:*



Az első webhely készen áll a hozzáféréshez.

FTP telepítése és beállítása

Az FTP (File Transfer Protocol) egy nagyon népszerű protokoll, amely lehetővé teszi a felhasználók számára a fájlok egyszerű feltöltését és letöltését.

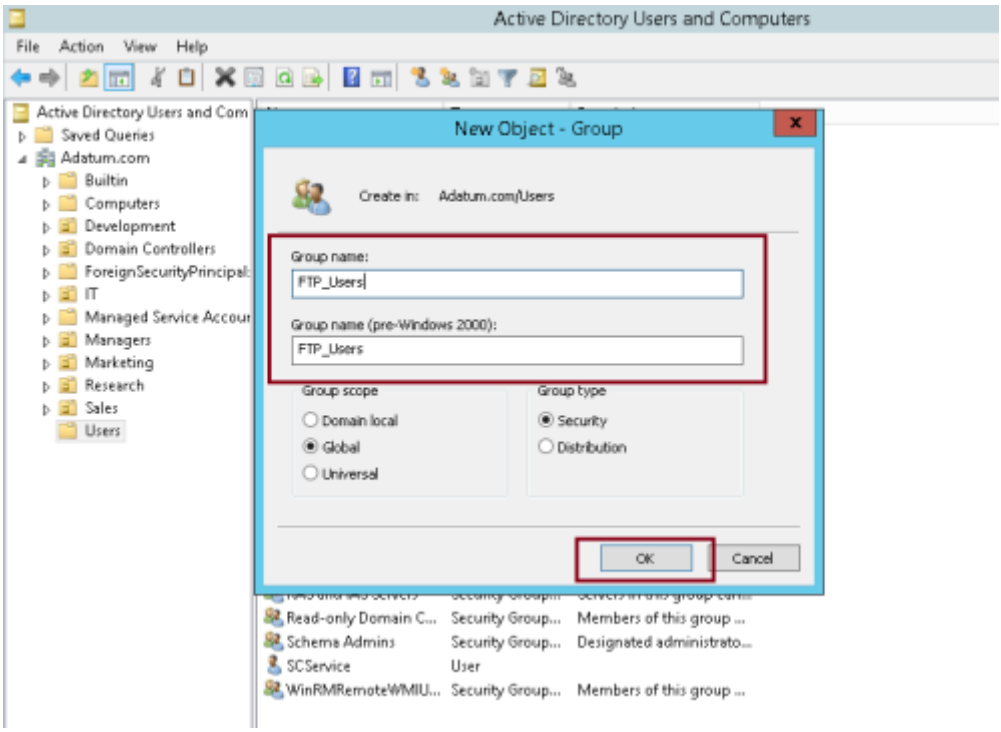
Még az FTP is tekintélyes adatátviteli technológiát tart, de még mindig használható és könnyen használható a Server Administrator részéről.

FTP-kiszolgálót konfigurálhat a Windows Server 2012 rendszerben az FTP-kiszolgáló szerepkör telepítésével.

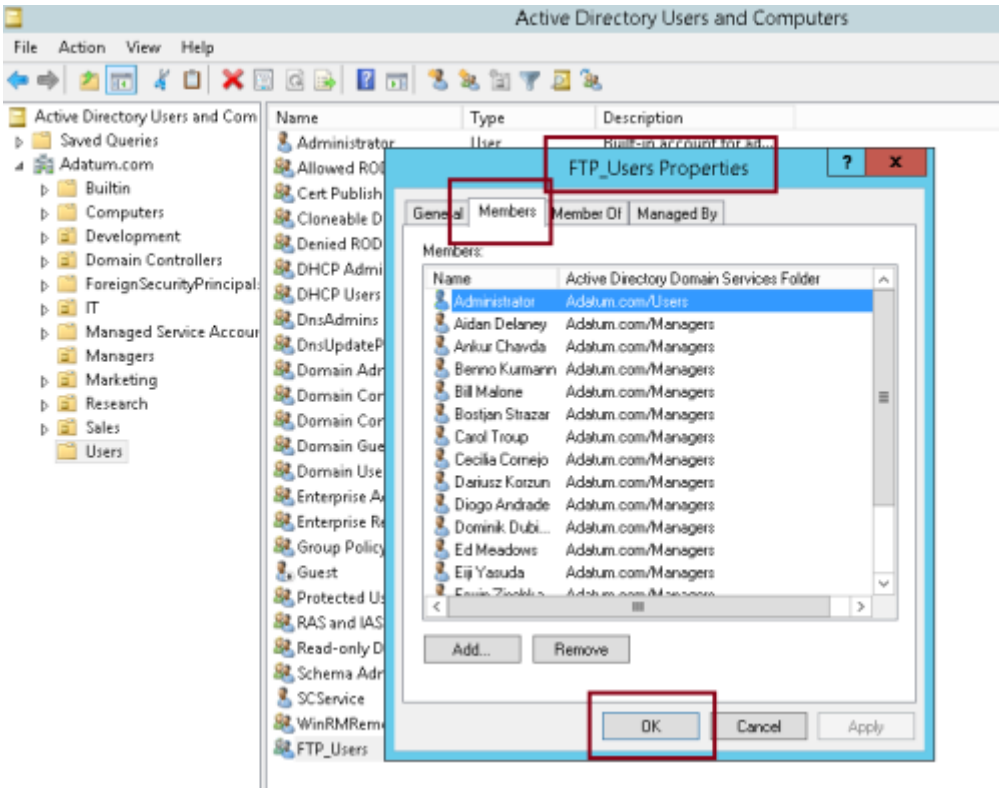
Ebben a bejegyzésben megmutatom neked egy nagyon egyszerű lépést az FTP kiszolgálói szerepkör telepítéséről és beállításáról a Windows Server 2012 R2 rendszerben.

1. - Az FTP-szerepkörök telepítése előtt meg kell adnia a hitelesítést a felhasználónak Domain környezetben.

1 - A Domain-kiszolgálón nyissa meg az Active Directory-felhasználókat és számítógépeket, és hozzon létre FTP_Users csoportot.

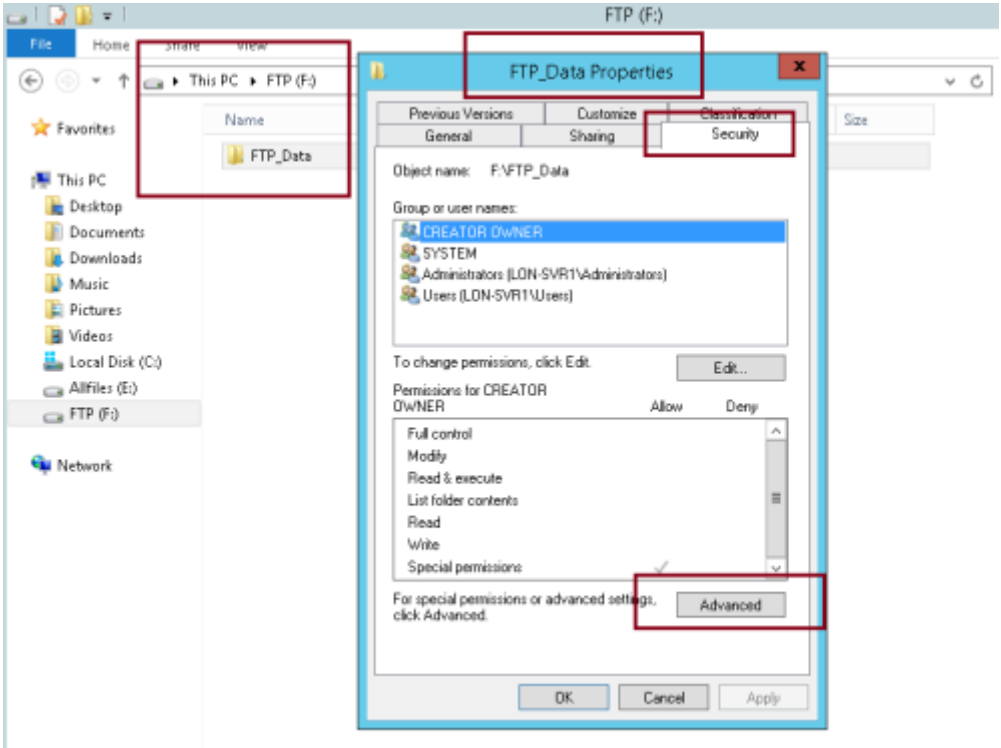


2 - Az FTP_Users tulajdonságoknál kérjük, add hozzá az Adminisztrátort, és minden olyan felhasználót, aki használni / bejelentkeznie kell az FTP kiszolgálóra ...



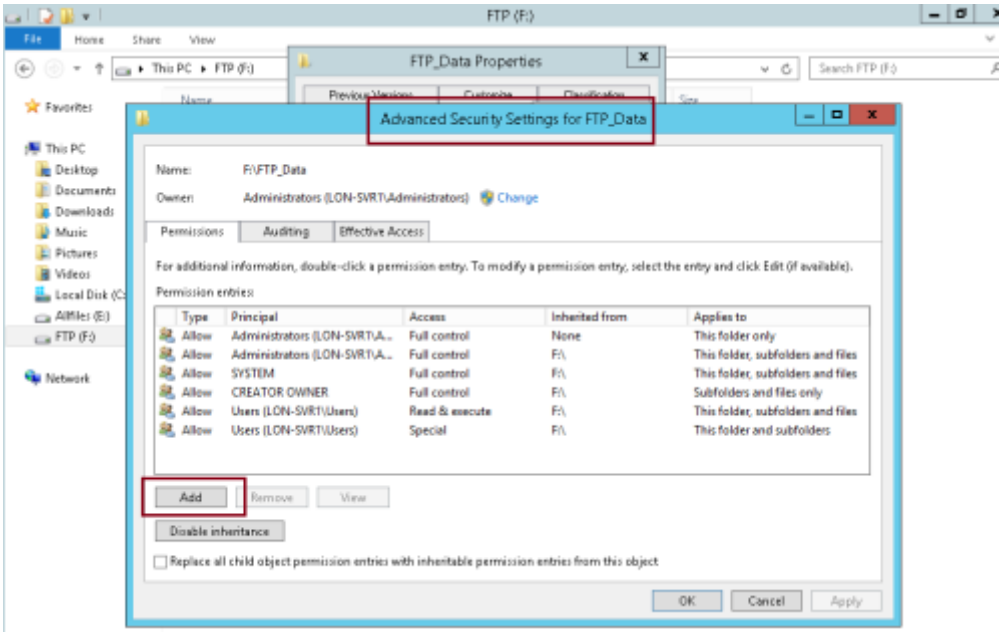
3 - Ezután váltson Member Serverre, és hozzon létre egy mappát az FTP hozzáféréshez, majd jobb gombbal kattintson az FTP mappára, és kattintson a Security (Biztonság) gombra.

** Az FTP mappa tulajdonságaiban, a Biztonság alatt kattintson a Speciális ... parancsra.

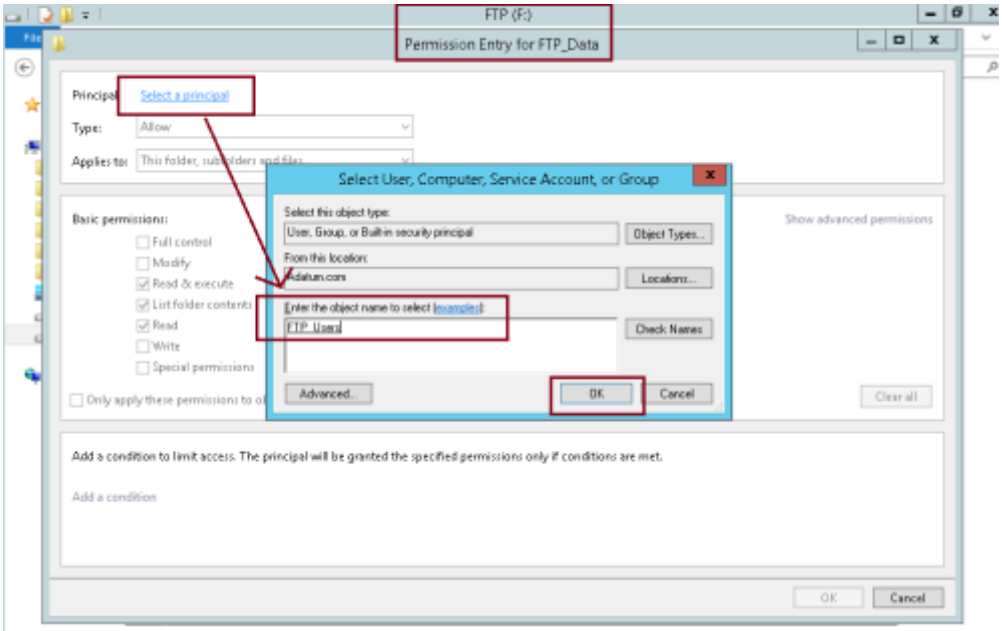


4 - Az FTP mappa részletes biztonsági beállításaihoz kattintson a Hozzáadás gombra.

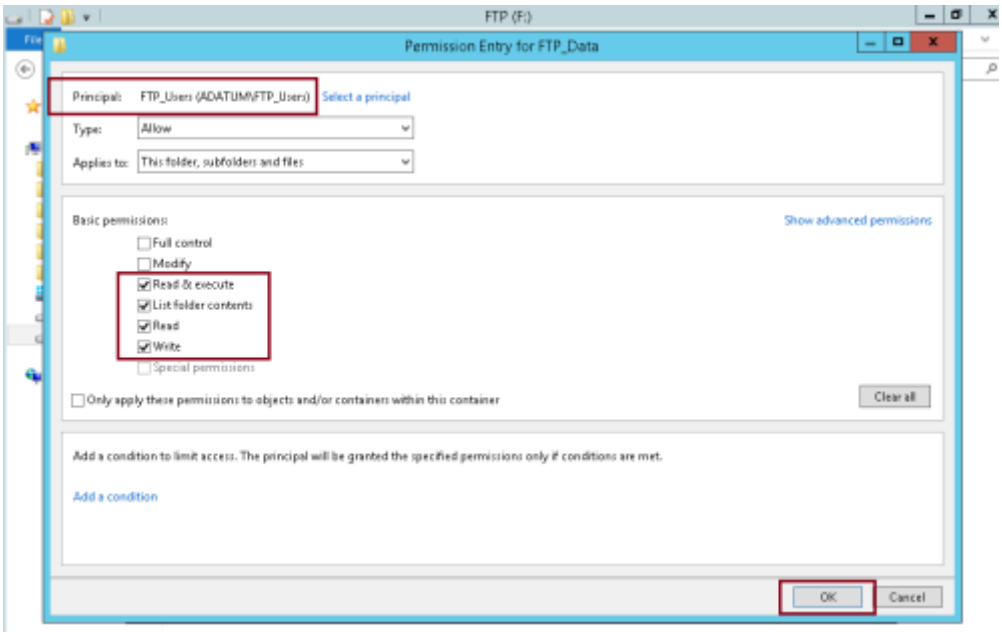
** Itt meg fogjuk adni az FTP_Users csoport hozzáférési jogosultságait.



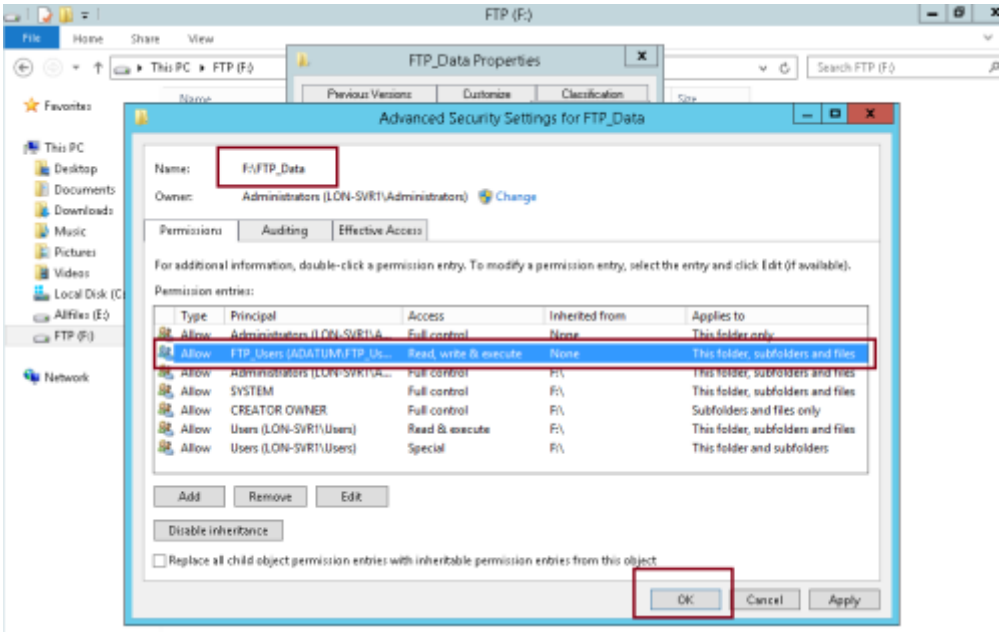
5 - Az FTP mappa engedélyezési bejegyzéséhez kattintson a fő link kiválasztása, majd a Beállítás megadása mezőbe írja be az FTP_Users parancsot, és kattintson az OK gombra.



6 - Az FTP mappa engedélyezési bejegyzéséhez az Alapvető jogosultságok alatt kattintson az Írás, majd az OK gombra.

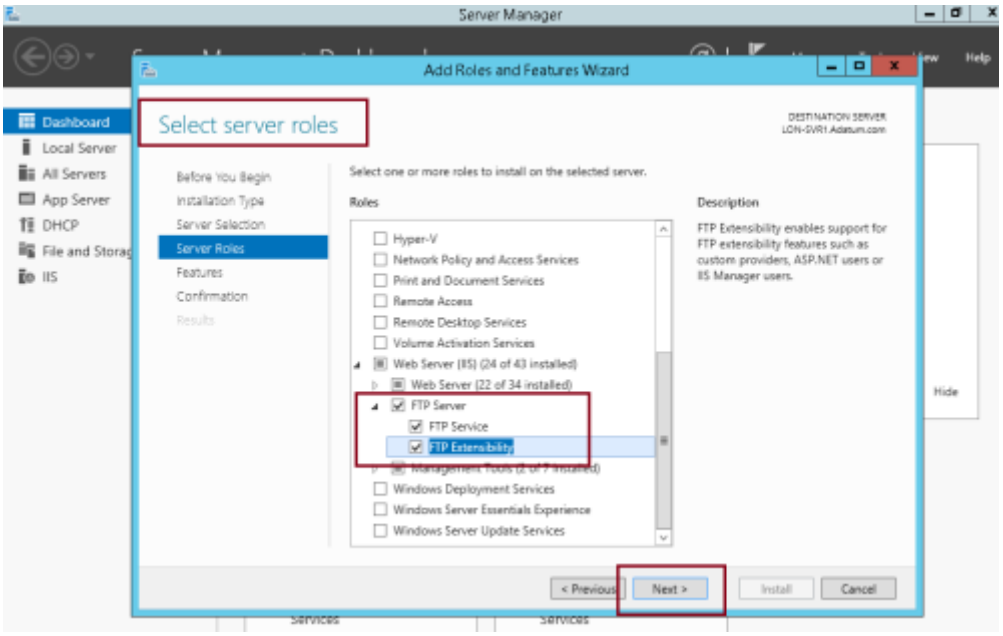


7 - ellenőrizze, hogy az FTP_Users szerepel-e az FTP mappában található Speciális biztonsági beállítások listáján ...

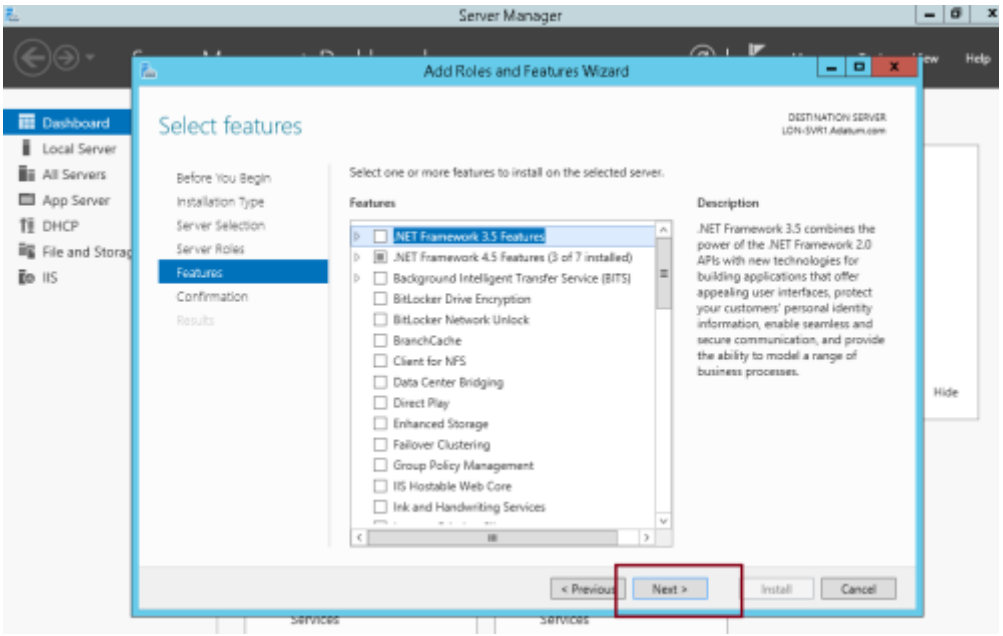


2. - FTP szerepkör szolgáltatás telepítése ...

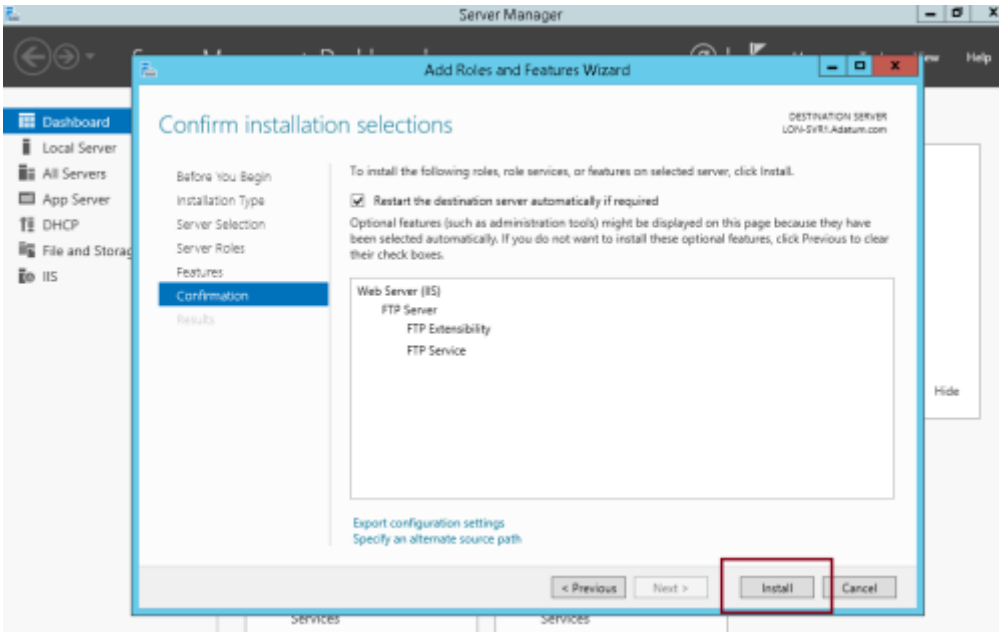
1 - Még mindig a tagkiszolgálón nyissa meg a Kiszolgálókezelőt, kattintson a Szerepkörök és szolgáltatások hozzáadása lehetőségre, majd folytassa a kiszolgálói szerepköröket, és keresse meg az FTP-kiszolgálót (kattintson az FTP-szolgáltatás és az FTP-bővíthetőség lehetőségre), majd kattintson a Tovább gombra.



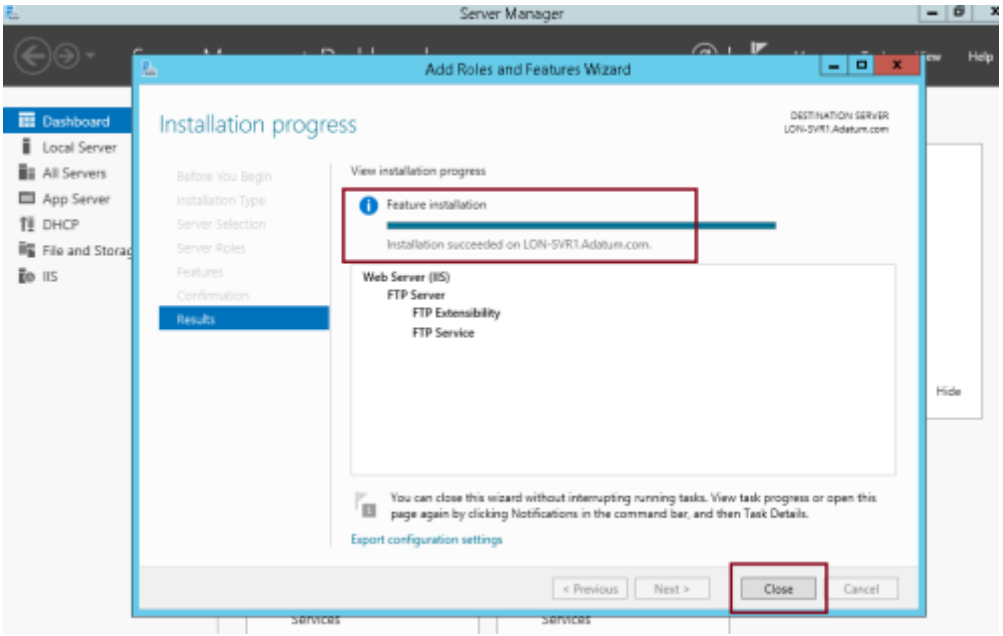
2 - A Válasszon funkciók felületen folytassa a Tovább ...



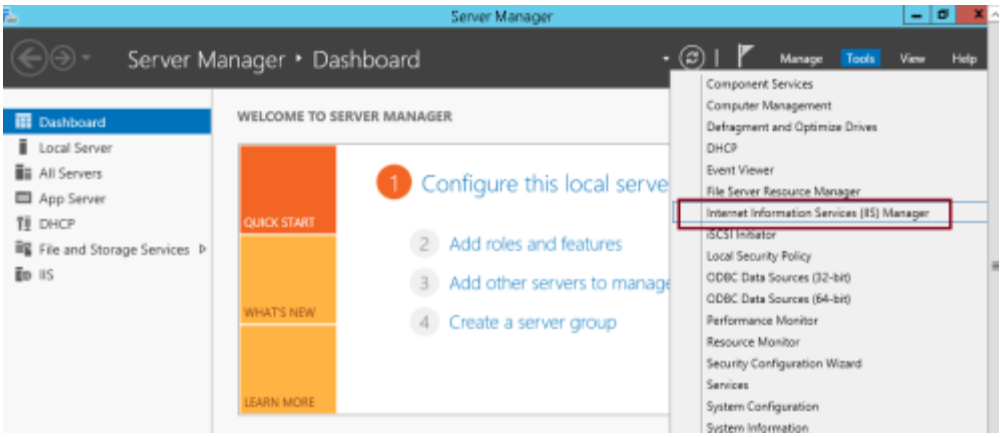
3 - A Telepítési beállítások megerősítése felületen kattintson a Telepítés ... gombra.



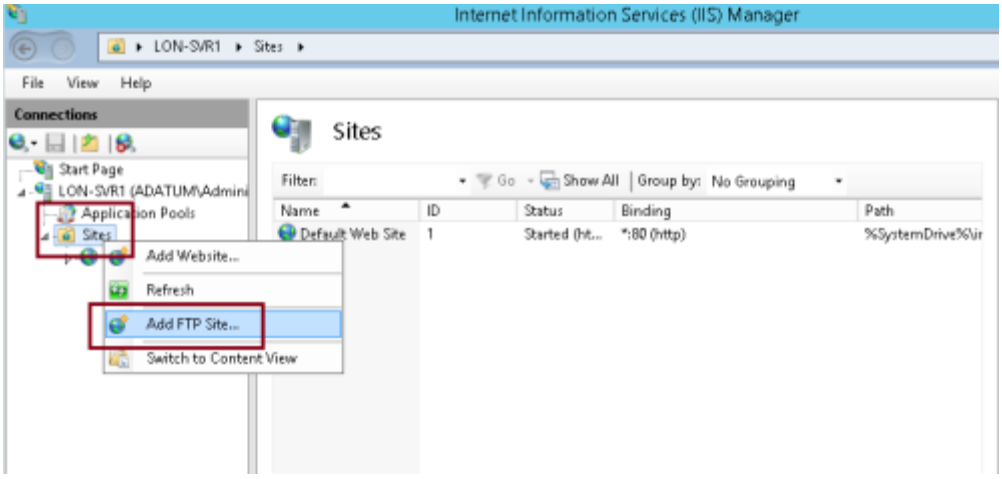
4 - Miután a telepítés befejeződött, indítsa újra a kiszolgálót ...



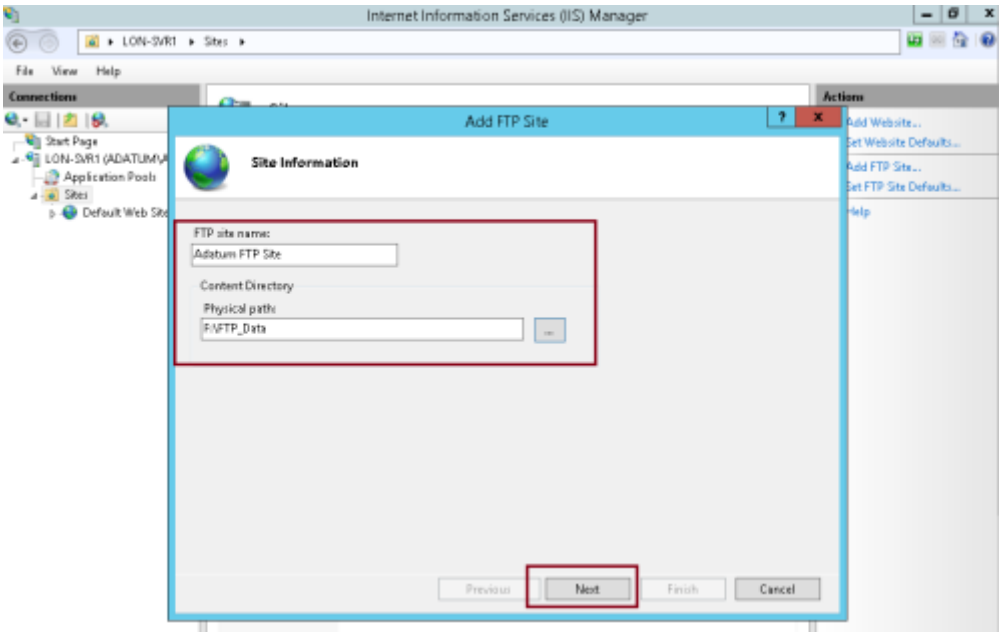
5 - Miután a szerver újraindult, nyissa meg a Kiszolgálókezelőt, kattintson az Eszközök gombra, és kattintson az Internet Information Services (IIS) kezelőre ...



6 - Az IIS konzolon kattintson jobb gombbal a Webhelyek, majd kattintson az FTP webhely hozzáadása lehetőségre.

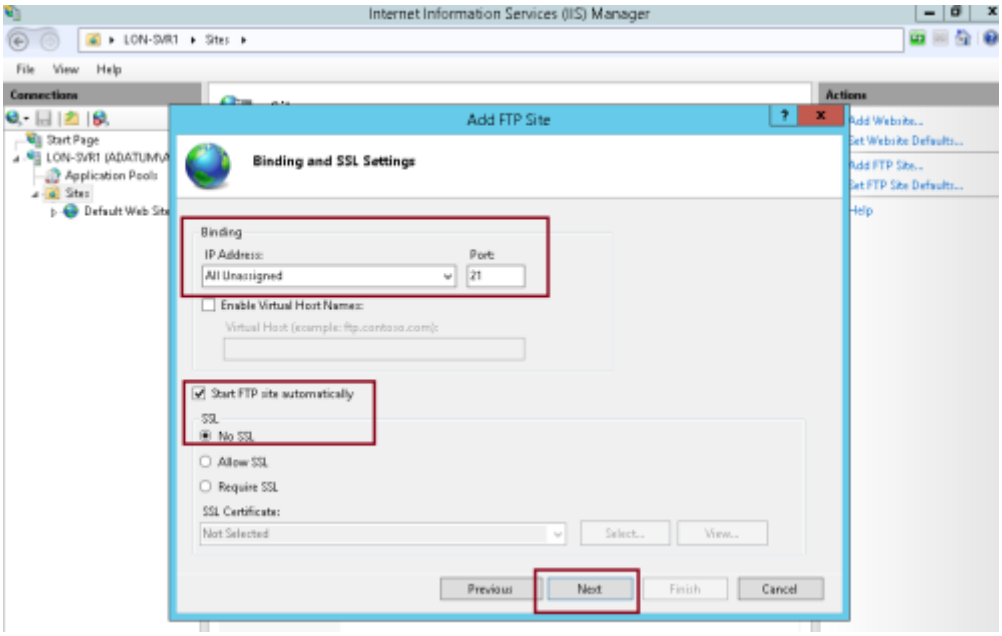


7 - Az FTP Site felületének hozzáadása az FTP webhely neve: mezőbe írja be a saját FTP-helynevét, majd a fizikai elérési utat, keresse meg az előző lépésben létrehozott FTP-mappát ...

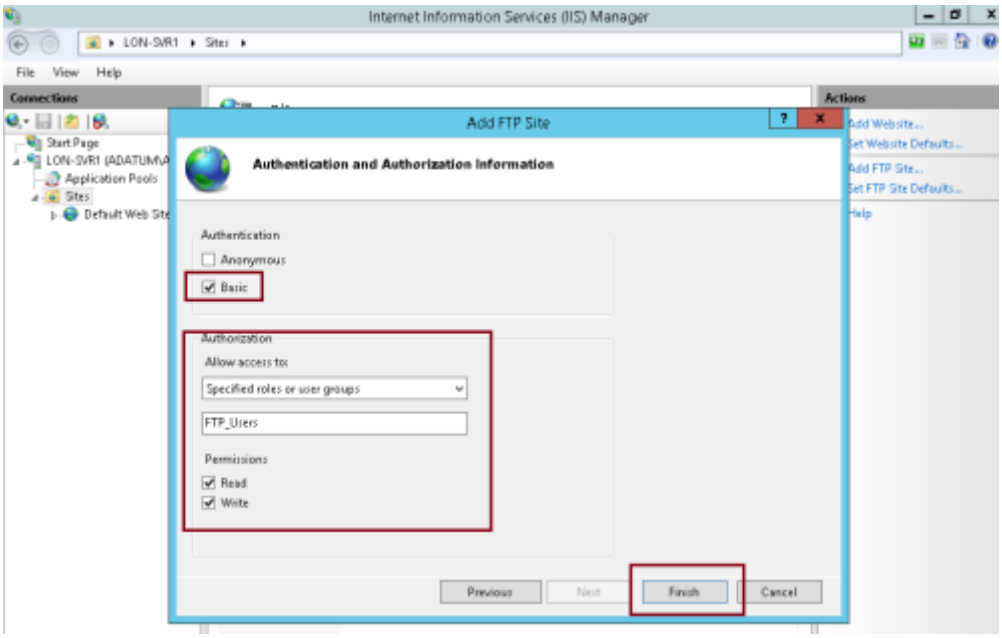


8 - Az FTP Site felület hozzáadása alatt ellenőrizze, hogy az összes hozzárendelt IP-cím a 21. porttal ...

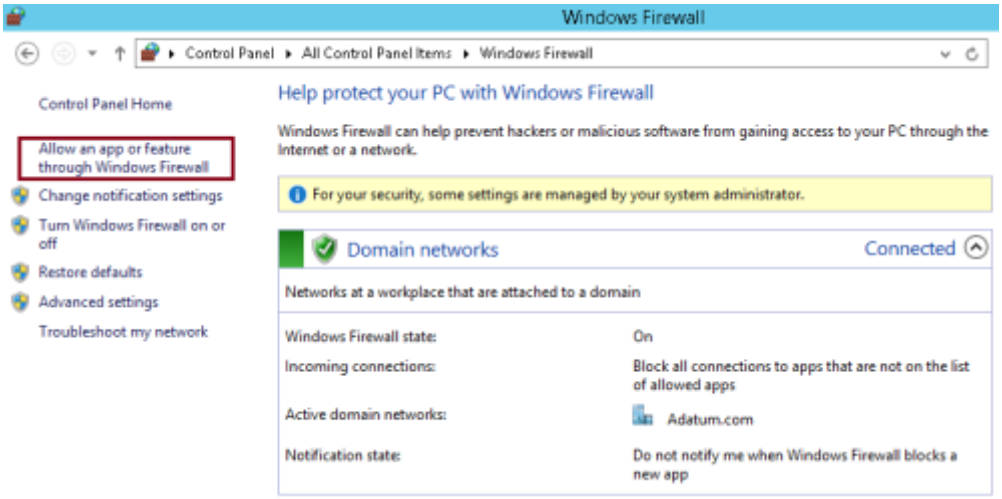
** kattintson a Nincs SSL (nem fogom biztosítani az internetes tanúsítványokkal rendelkező FTP-helyet) ... és kattints a Tovább gombra.



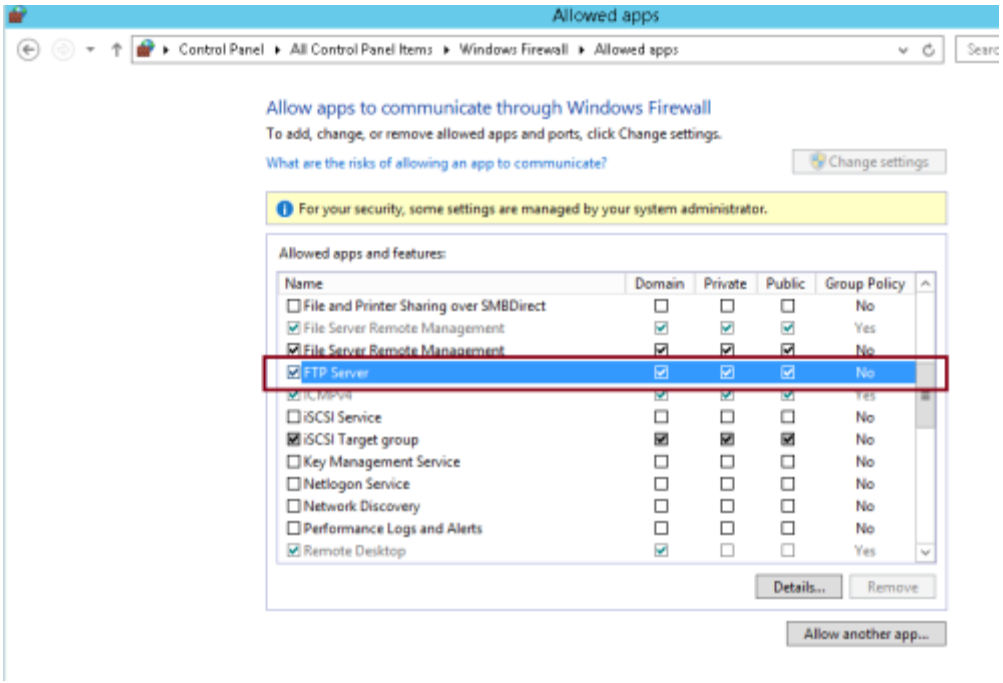
9 - A Hitelesítés alatt kattintson az Alap, a Hozzáférés engedélyezése menüpont alatt, válassza a Meghatározott szerepek vagy felhasználói csoportok parancsot, írja be az FTP_Users parancsot, ellenőrizze, hogy az Engedélyek alatt kattintson az Olvasás és írás lehetőségre, majd kattintson a Befejezés ... gombra.



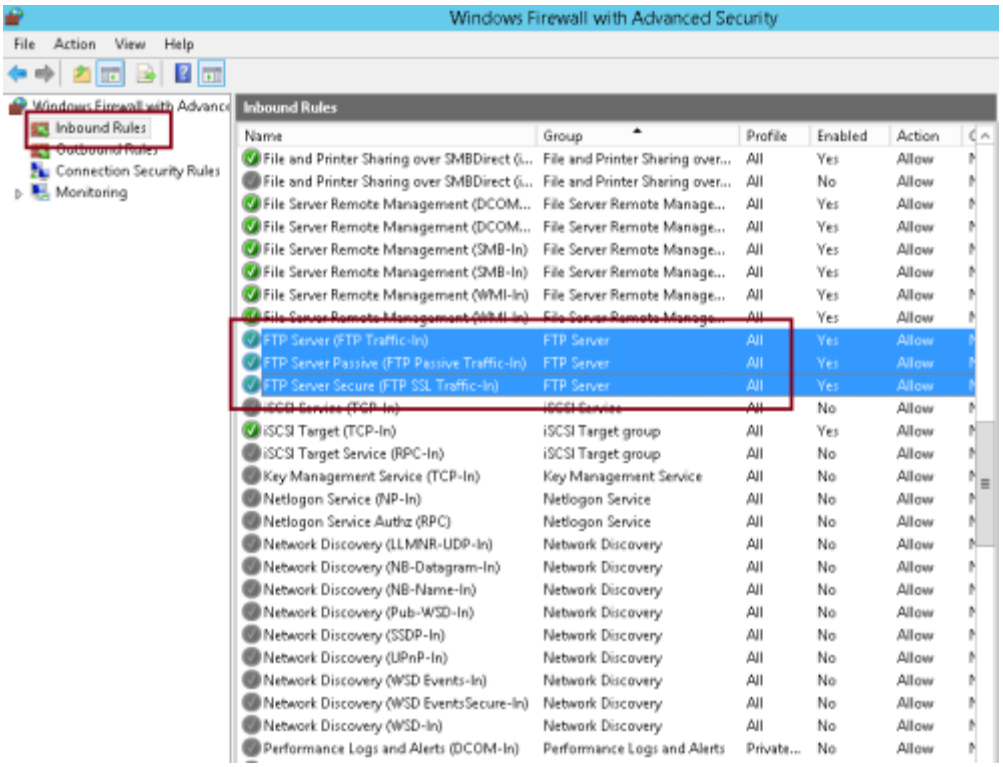
10 - Ezután nyissa meg a Windows tűzfalat, majd kattintson a Hozzáférés vagy szolgáltatás engedélyezése a Windows tűzfalon keresztül ...



11 - Az Alkalmazások engedélyezése a Windows tűzfalon keresztül történő kommunikáció során keresse meg az FTP-kiszolgálót, és ellenőrizze, hogy a Domain, a Privát és a nyilvános jelölőnégyzet be van-e jelölve, majd kattintson az OK gombra.



12 - Ezután nyissa meg a Windows tűzfalat a Speciális biztonsággal, kattintson a Bejövő szabályok elemre, és görgessen az FTP-re (ellenőrizze, hogy a 3 FTP-összetevő szerepel-e) ...

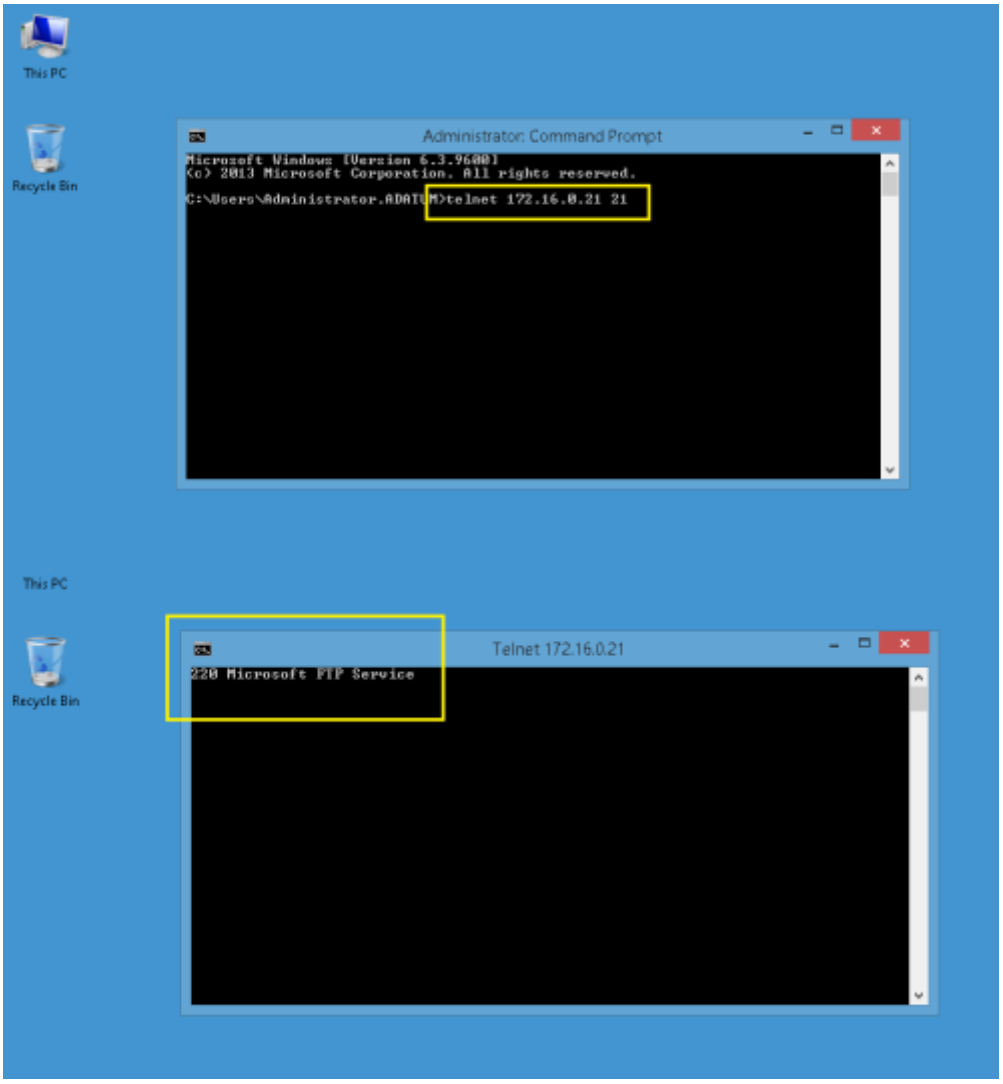


13 - Most átkapcsolhat az ügyfélszámítógépre (ebben a bemutatóban a Windows 8.1-et használom) ...

** az ügyfélszámítógépen, nyissa meg a CMD és a type telnet 172.16.0.21 21 fájlt

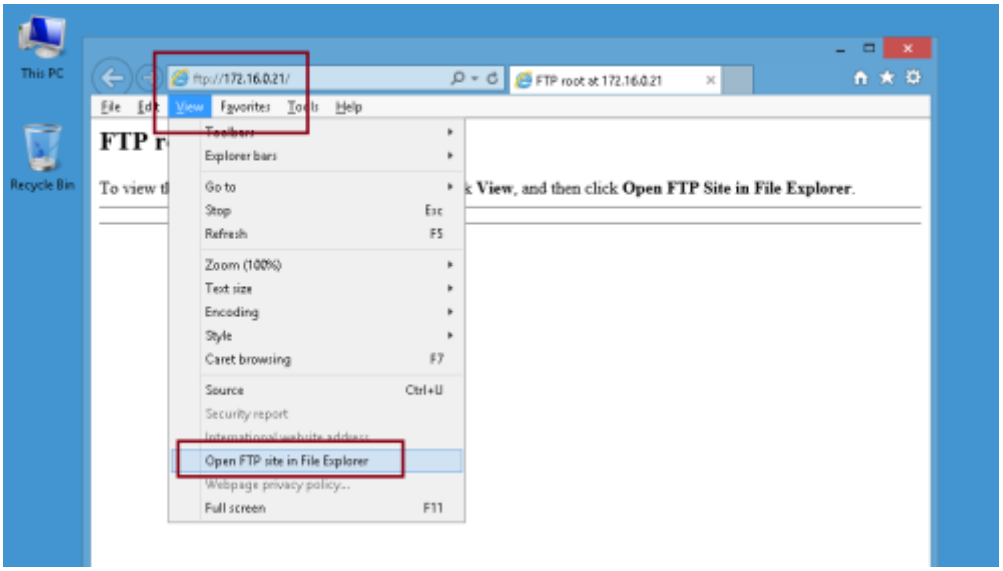
** 172.16.0.21 -> FTP szerver IP

** 21 -> FTP port száma



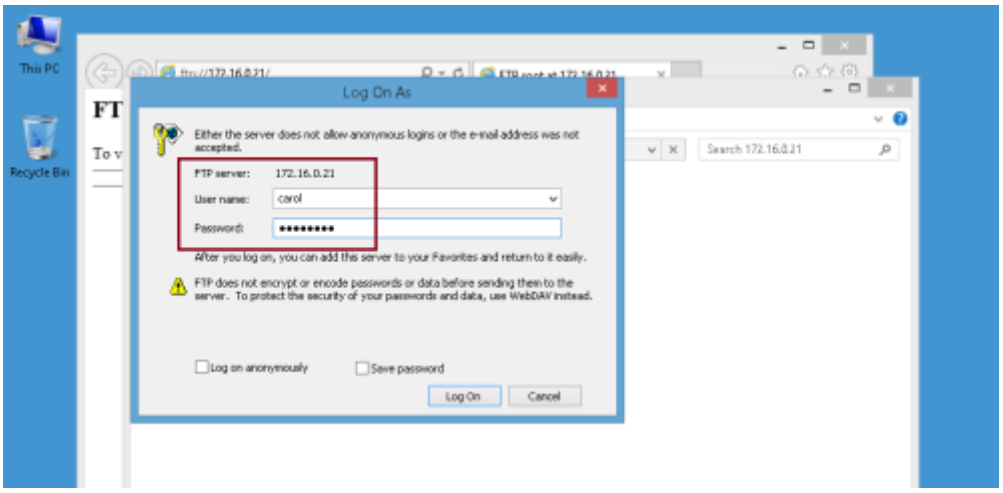
14 - a CMD-ben, azt állította 220 Microsoft FTP Service (sikeresen csatlakozik az FTP szerverhez) ...

15 - Nyissuk meg a webböngészőt a ftp://172.16.0.21 címsávban és írjuk be, majd kattintsunk a View menüre, és kattintsunk az FTP site megnyitása a File Explorer-ben ...

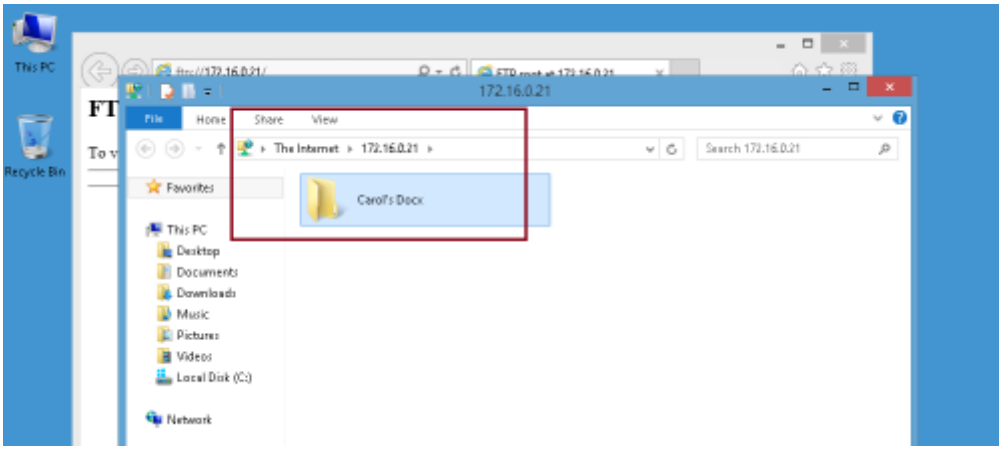


16 - A Bejelentkezés állapotában

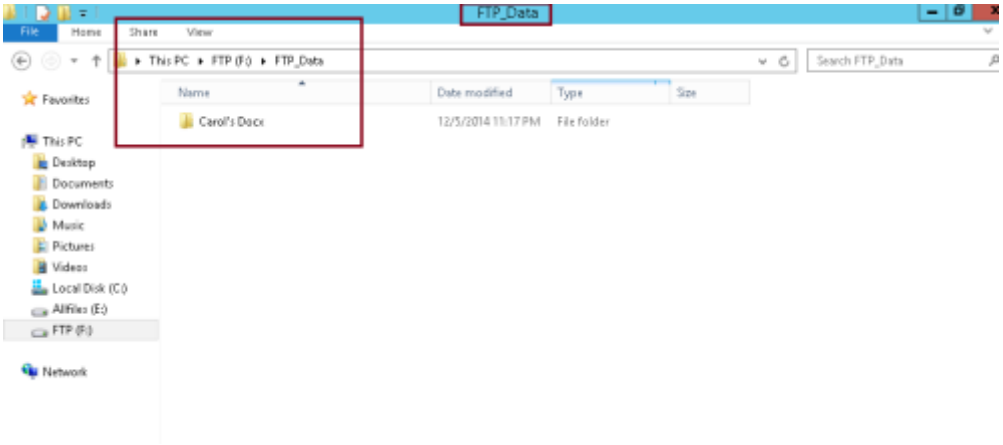
jelölőnégyzetet felhasználónévvel töltse ki az összes FTP_Users felhasználót, majd töltse ki a jelszót és kattintson a Bejelentkezés ... gombra.



17 - Miután sikeresen bejelentkezett az FTP szerverre, megpróbálhat létrehozni bármilyen mappát ...



18 - Végül térjen vissza az FTP szerverre, és erősítse meg, hogy a létrehozott mappa szerepel az FTP kiszolgálón ...



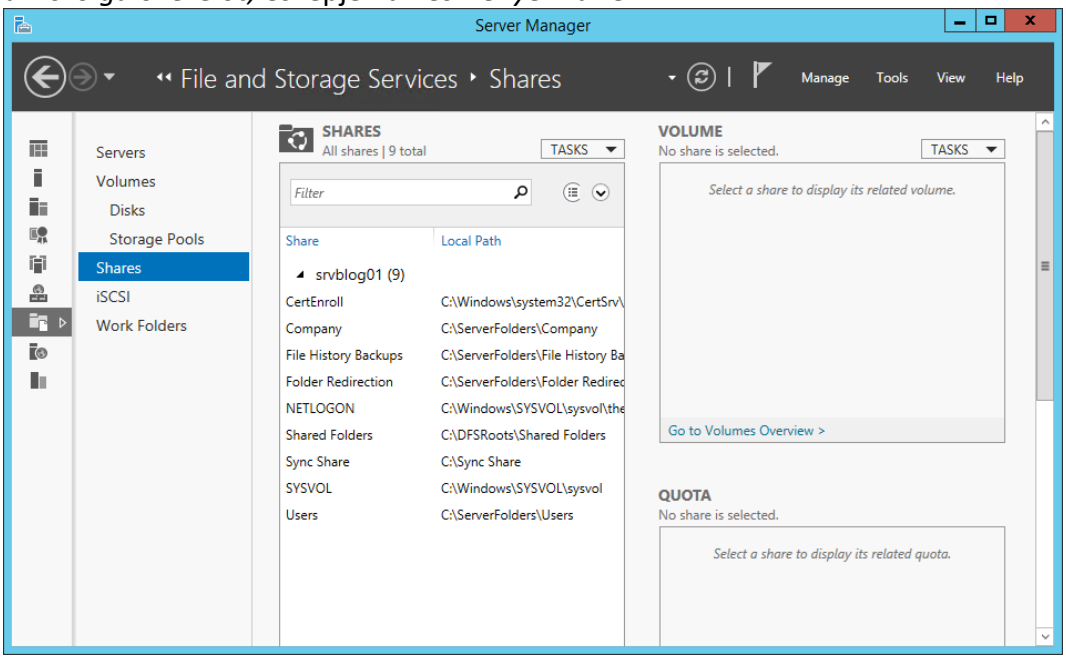
Romaing profil készítése

A barangolási profilok lehetővé teszik az Active Directory tartományának felhasználók számára, hogy hozzáférjenek az asztalukhoz és a dokumentumokhoz a tartomány bármelyik számítógépről.

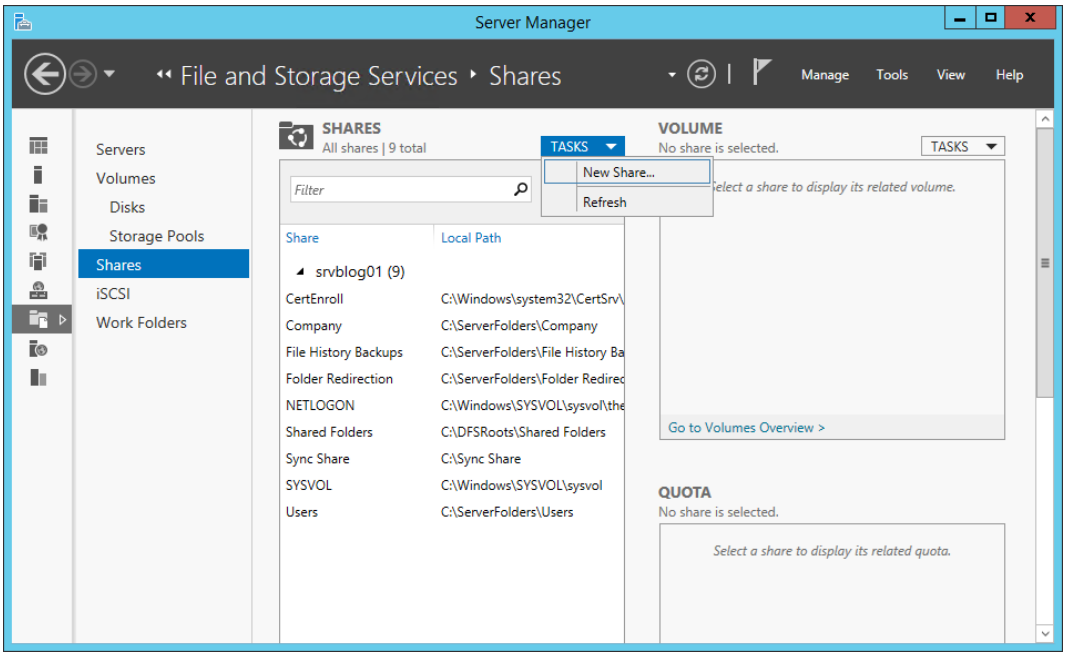
Ez egy olyan erőteljes szolgáltatás, amely javíthatja az alkalmazottak termelékenységét és megkönnyítheti életét.

A legjobb a **Roaming profiloknál** , hogy könnyű beállítani őket.

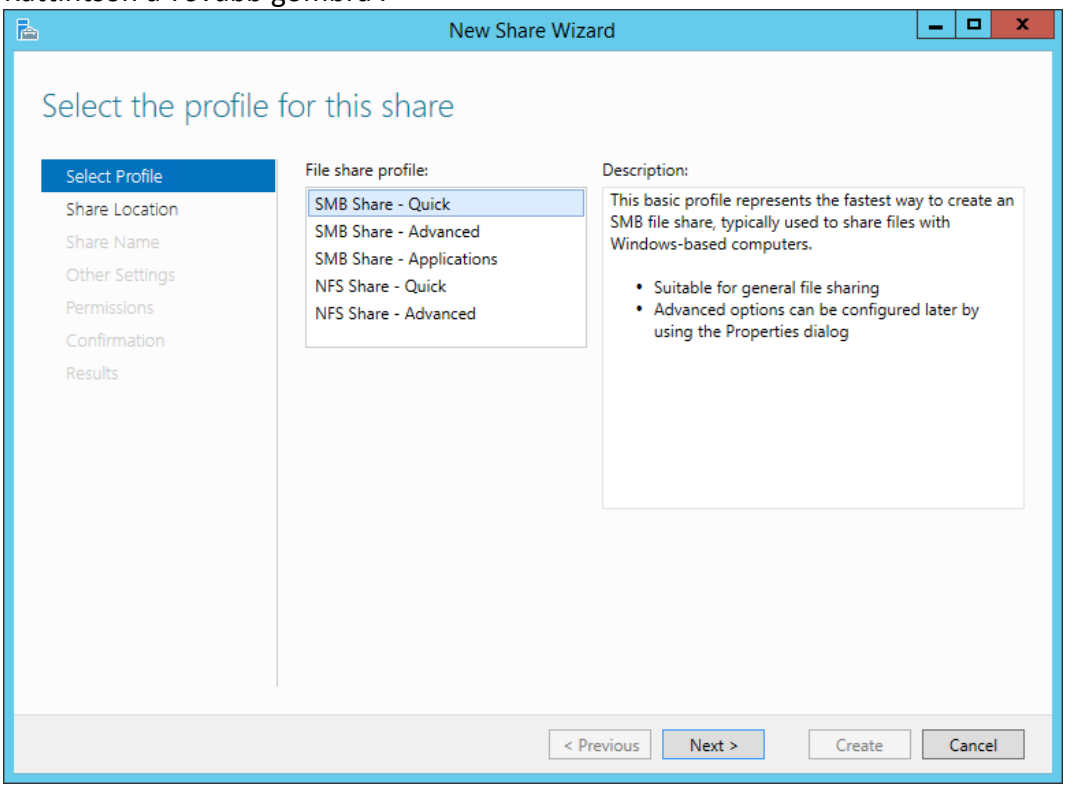
A **roamingprofil** beállítása előtt létre kell hoznunk egy *megosztást* . Nyissa meg a *Kiszolgálókezelőt,* és lépjen a *Részvények* fülre:



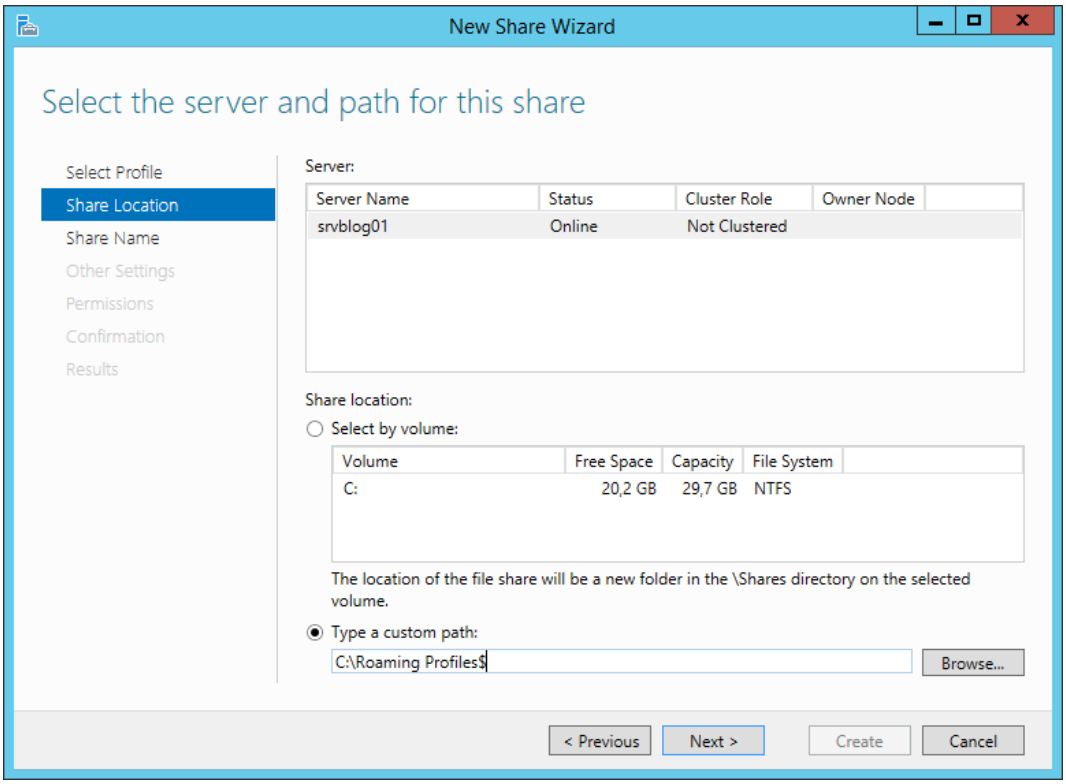
Új *megosztás* létrehozása *:*



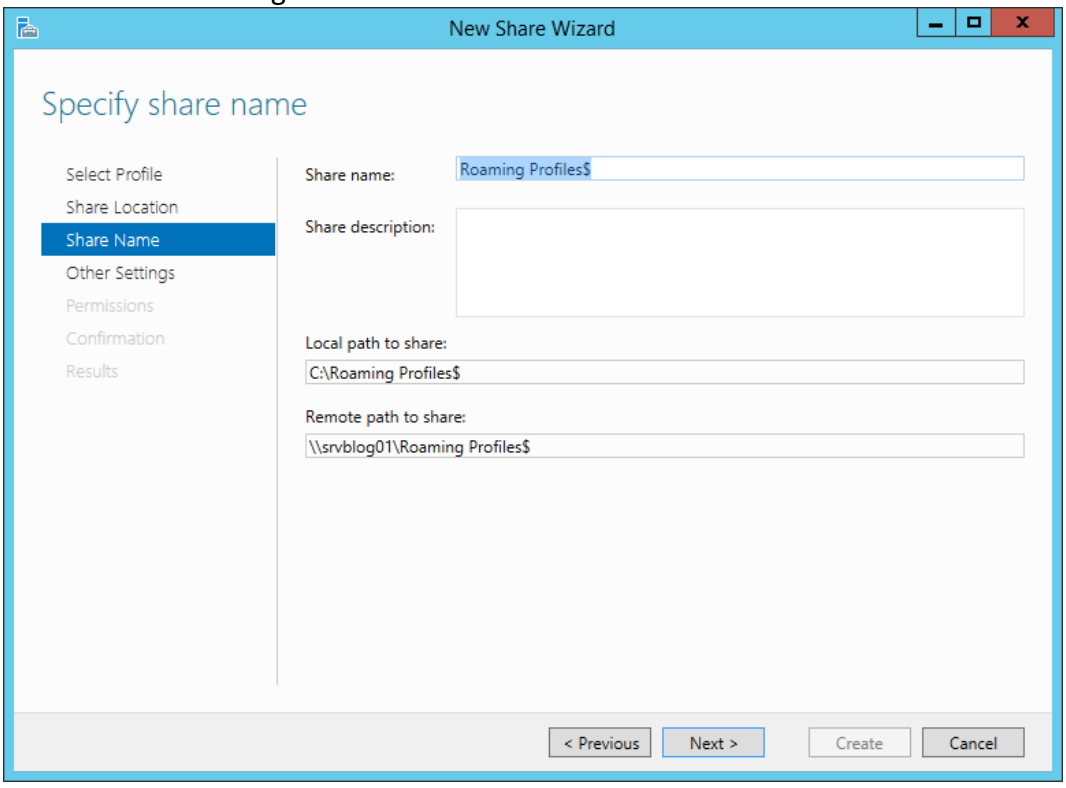
Kattintson a *Tovább* gombra *:*



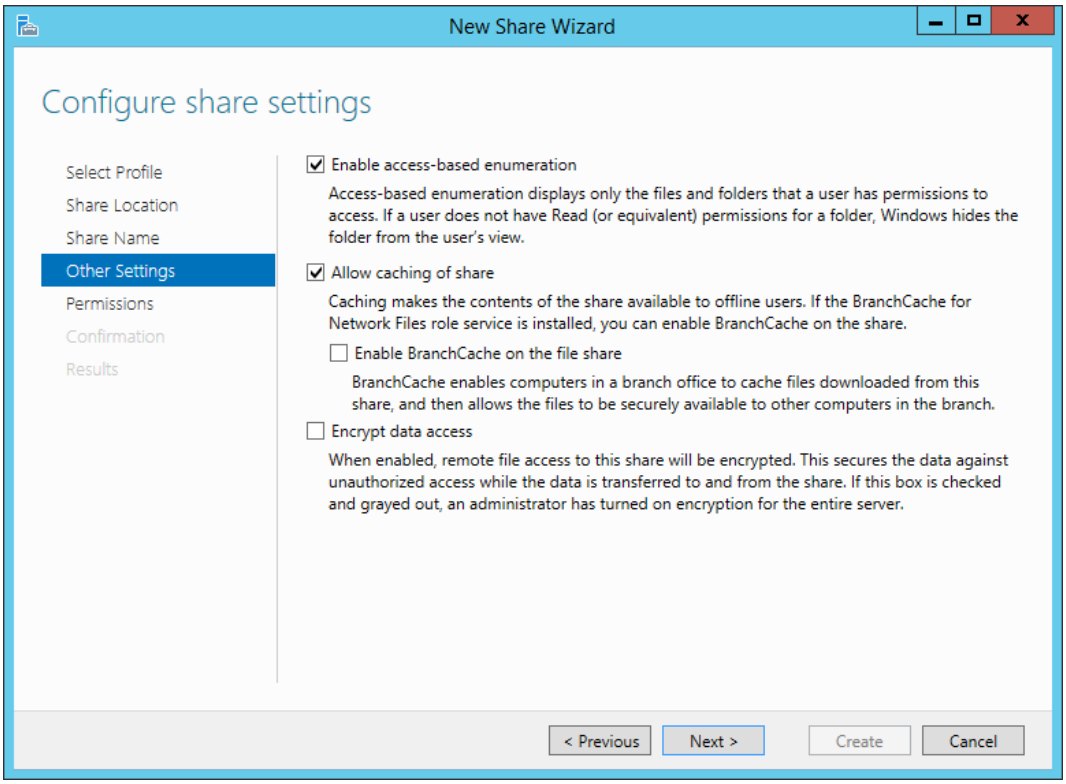
Adja meg a *barangolási profilok* elérési útját . A megosztott mappát láthatatlanná tegye egy *$* -ot az elérési út vége felé:



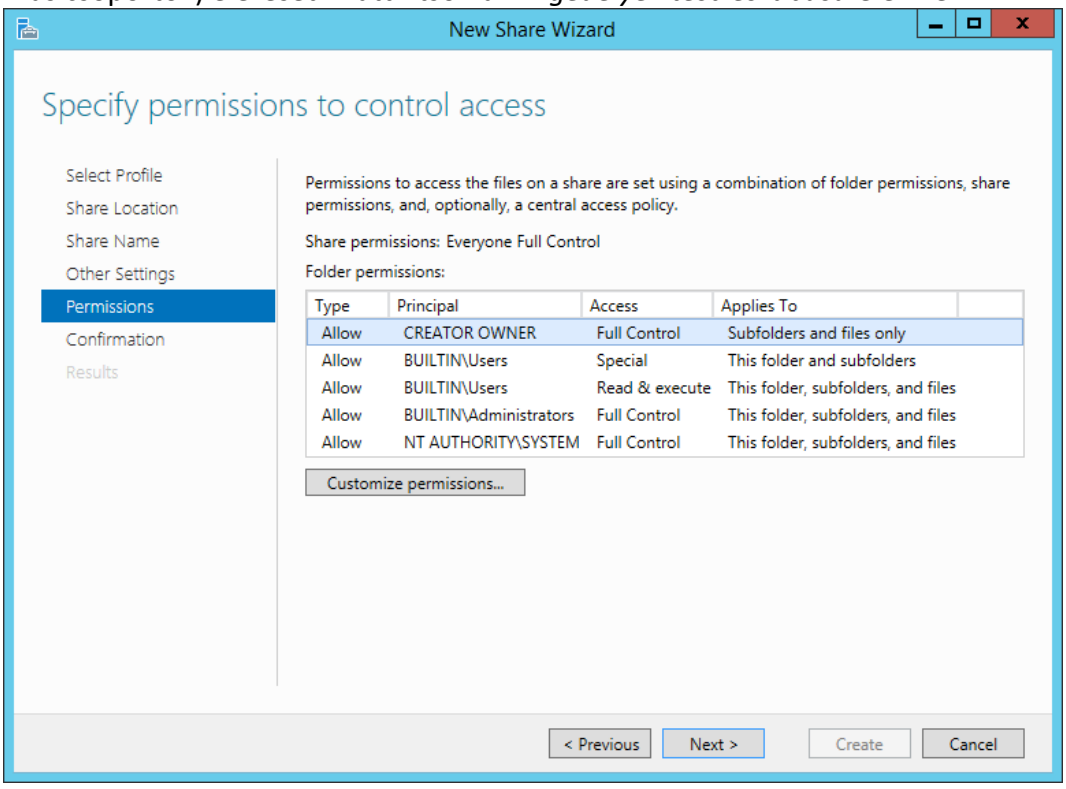
Kattintson a *Tovább* gombra *:*



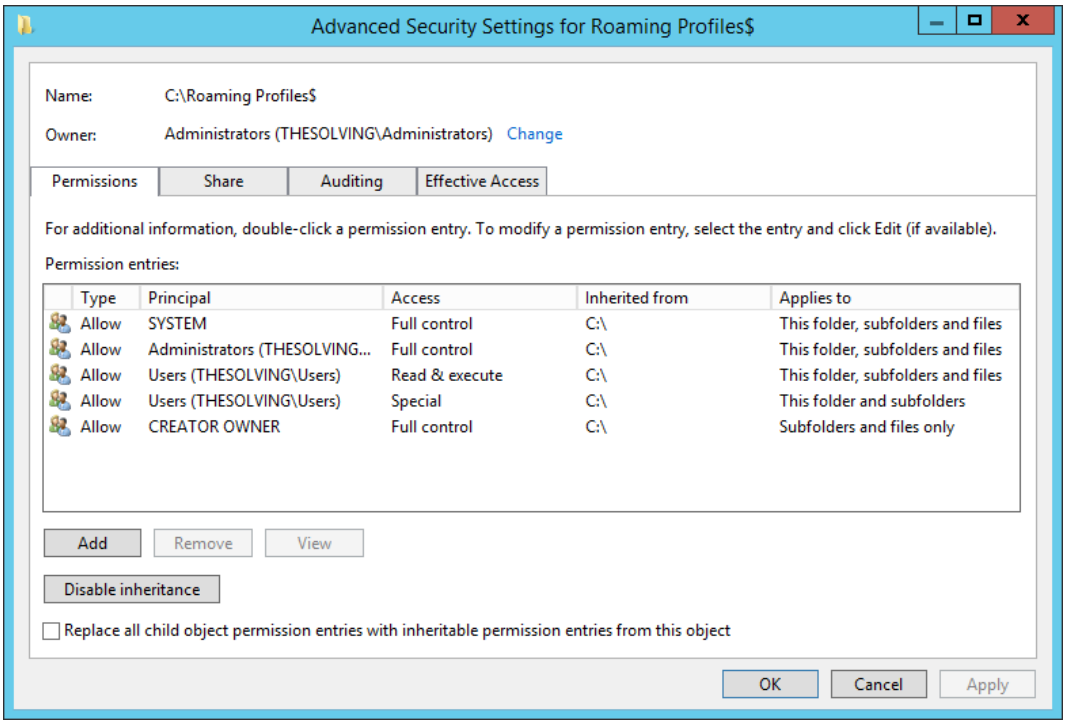
A *hozzáférési alapú számlálás* engedélyezése (a jobb biztonság érdekében):



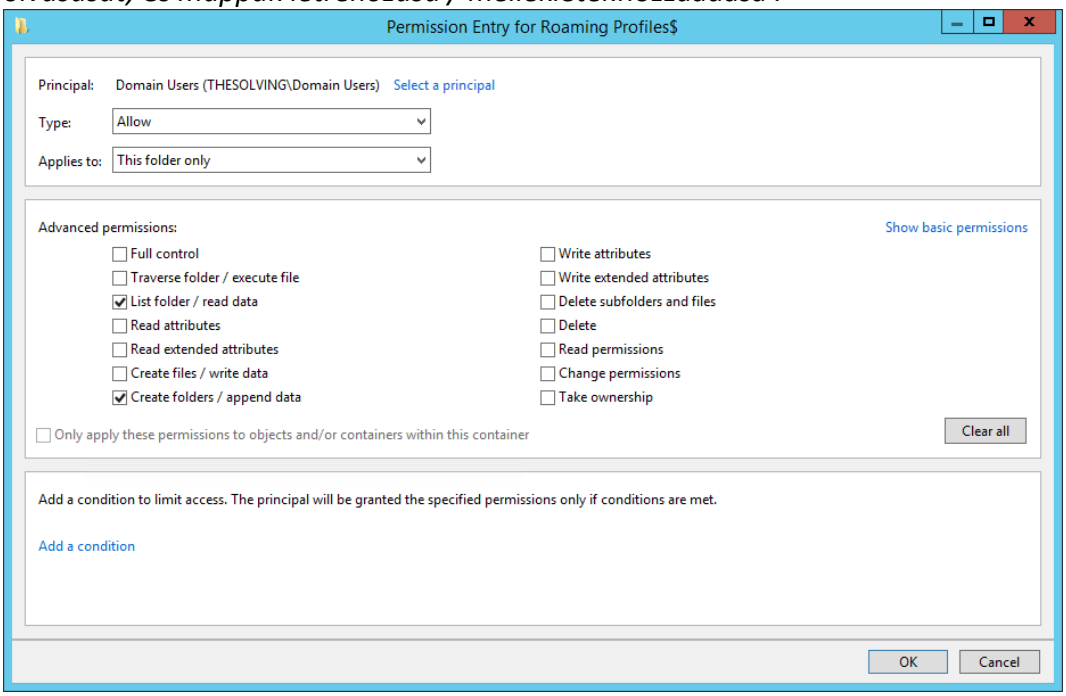
Itt az ideje, hogy testreszabja az engedélyeket. Engedélyezzük a *domain felhasználók* (vagy más csoportok) elérését . Kattintson az *Engedélyek testreszabása* elemre *:*



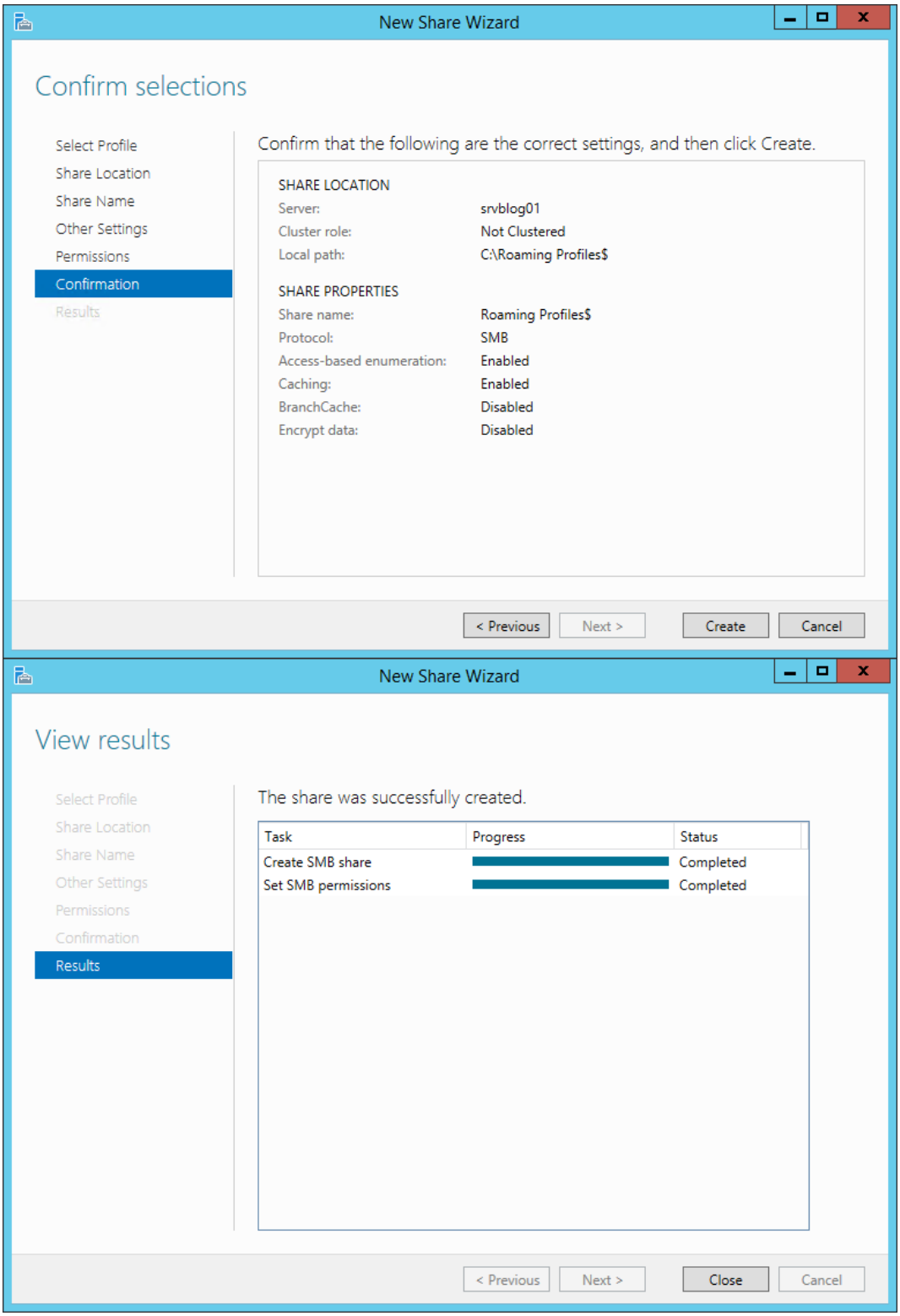
Kattintson a *Hozzáadás* gombra *:*



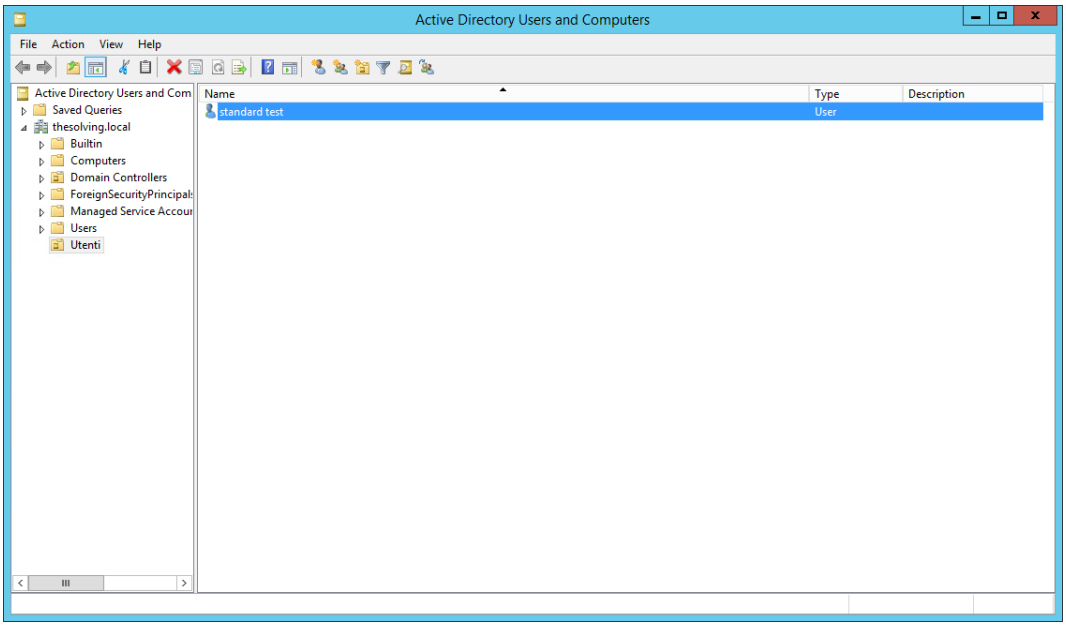
Válassza ki a csoportot (*Domain felhasználók* a példánkban), és alkalmazza az engedélyeket *csak* erre a *mappára* . Engedélyeznie kell a *Lista mappákat / az adatok olvasását,* és *mappák létrehozása / mellékletek*hozzáadása *:*



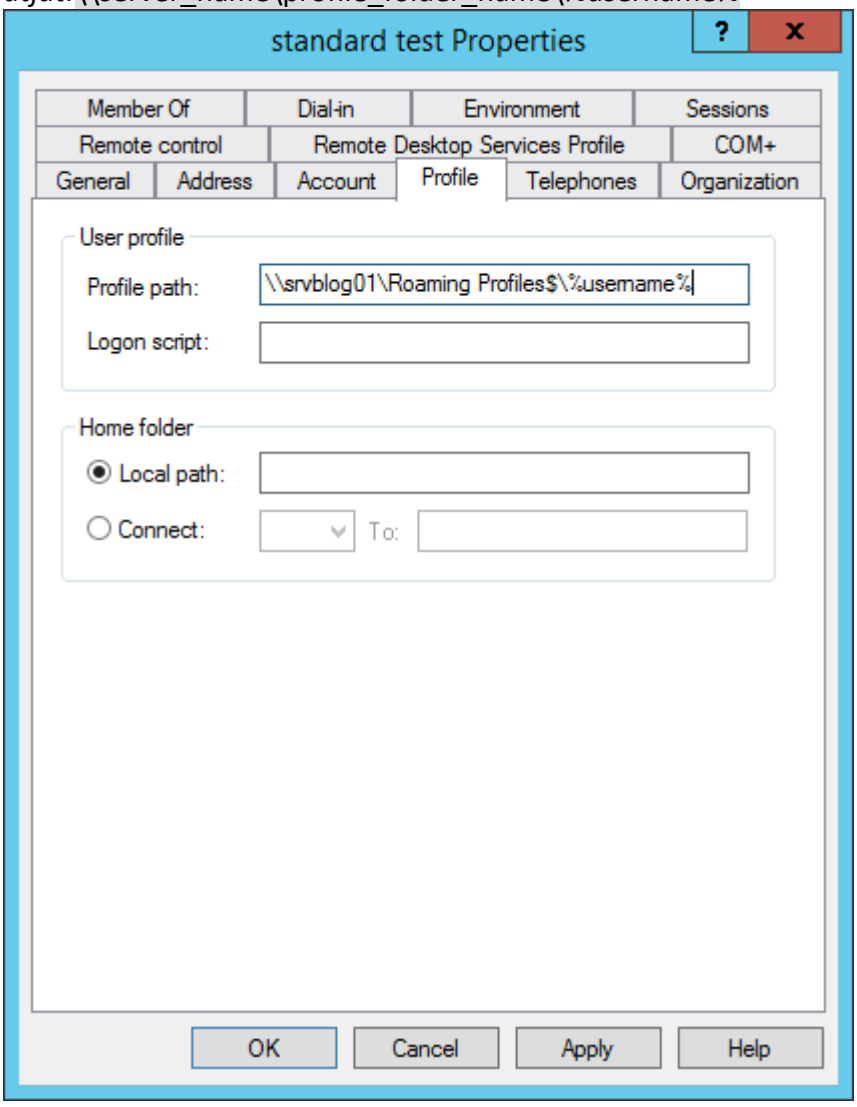
Erősítse meg, és a megosztás létrejön:



Most az utolsó lépés. Nyissa meg az *Active Directory felhasználók és számítógépek* panelet:



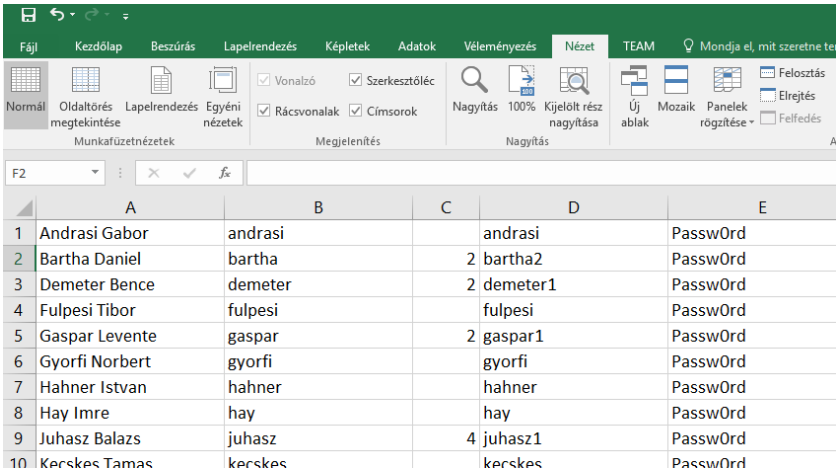
Nyissa meg a felhasználó *tulajdonságait* , és lépjen a *Profil* fülre. Adja meg a *profil elérési útját:*\\server_name\profile_folder_name\%username%



Küldetés teljesítve! Az első **barangolási profilja** aktív és aktív. Más **Roaming-profilok** létrehozásáhozhasználja a másolási funkciókat, vagy manuálisan adja meg a *Profil útvonalat* .
Szabályt is létrehozhat a folyamat automatizálására.

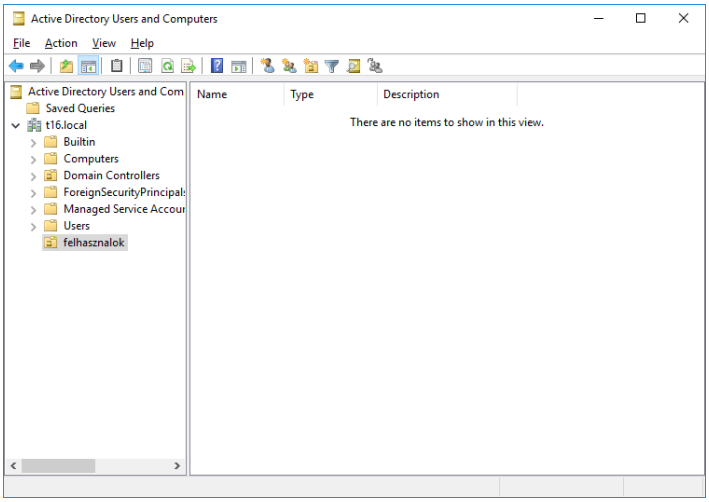
Sok felhasználsó létrehozása egyszerre

1. A nevekből leszedem a vezetékneveket kisbetüvel
   =KISBETŰ(BAL(A1;(SZÖVEG.KERES(" ";A1))-1))
2. Megnézem, melyikből van több
   =HA(DARABTELI(B1:$B$550;B1)>1;DARABTELI(B1:$B$550;B1);"")
3. Összefűzöm a neveked a darabszámokkal, ha több van, mint 1 db
   =ÖSSZEFŰZ(KISBETŰ(BAL(A1;(SZÖVEG.KERES(" ";A1))-1))
   ;HA(DARABTELI(B1:$B$550;B1)>1;DARABTELI(B2:$B$550;B2);""))

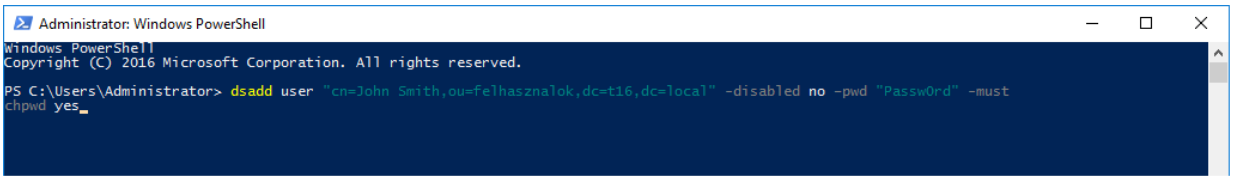


	A	B	C	D	E
1	Andrasi Gabor	andrasi		andrasi	Passw0rd
2	Bartha Daniel	bartha	2	bartha2	Passw0rd
3	Demeter Bence	demeter	2	demeter1	Passw0rd
4	Fulpesi Tibor	fulpesi		fulpesi	Passw0rd
5	Gaspar Levente	gaspar	2	gaspar1	Passw0rd
6	Gyorfi Norbert	gyorfi		gyorfi	Passw0rd
7	Hahner Istvan	hahner		hahner	Passw0rd
8	Hay Imre	hay		hay	Passw0rd
9	Juhasz Balazs	juhasz	4	juhasz1	Passw0rd
10	Kecskes Tamas	kecskes		kecskes	Passw0rd

4. Létrehozom a szervezeti egységet

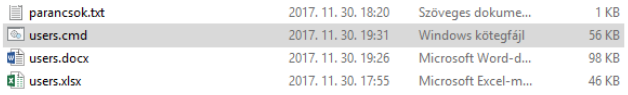


5. Lepróbálom a parancsot
   dsadd user "cn=John Smith,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "Passw0rd" -mustchpwd yes

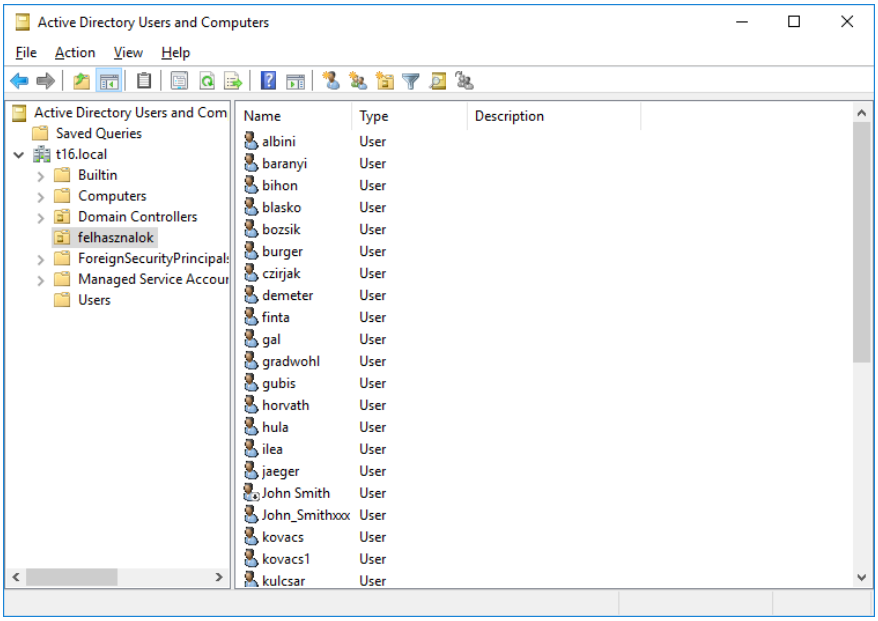


Elkészítem a .cmd fájl-t

dsadd user "cn=	andrasi	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	bartha2	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	demeter1	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	fulpesi	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	gaspar1	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	gyorfi	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	hahner	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	hay	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	juhasz1	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes
dsadd user "cn=	kecskes	,ou=felhasznalok,dc=t16,dc=local" -disabled no -pwd "	Passw0rd	" -mustchpwd yes



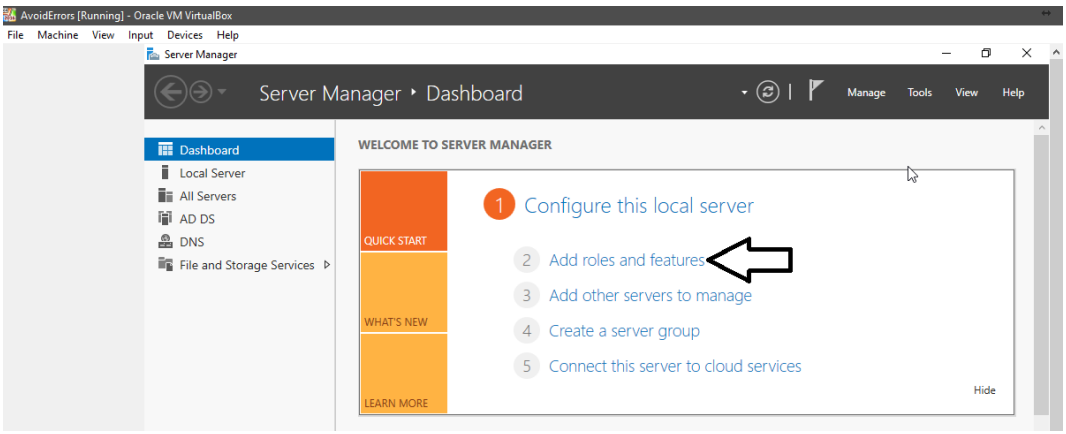
Tádáá


Active Directory Users and Computers
File Action View Help
Active Directory Users and Com
Saved Queries
t16.local
Builtin
Computers
Domain Controllers
felhasznalok
ForeignSecurityPrincipals
Managed Service Accour
Users
Name Type Description
albini User
baranyi User
bihon User
blasko User
bozsik User
burger User
czirjak User
demeter User
finta User
gal User
gradwohl User
gubis User
horvath User
hula User
ilea User
jaeger User
John Smith User
John_Smithxxx User
kovacs User
kovacs1 User
kulcsar User

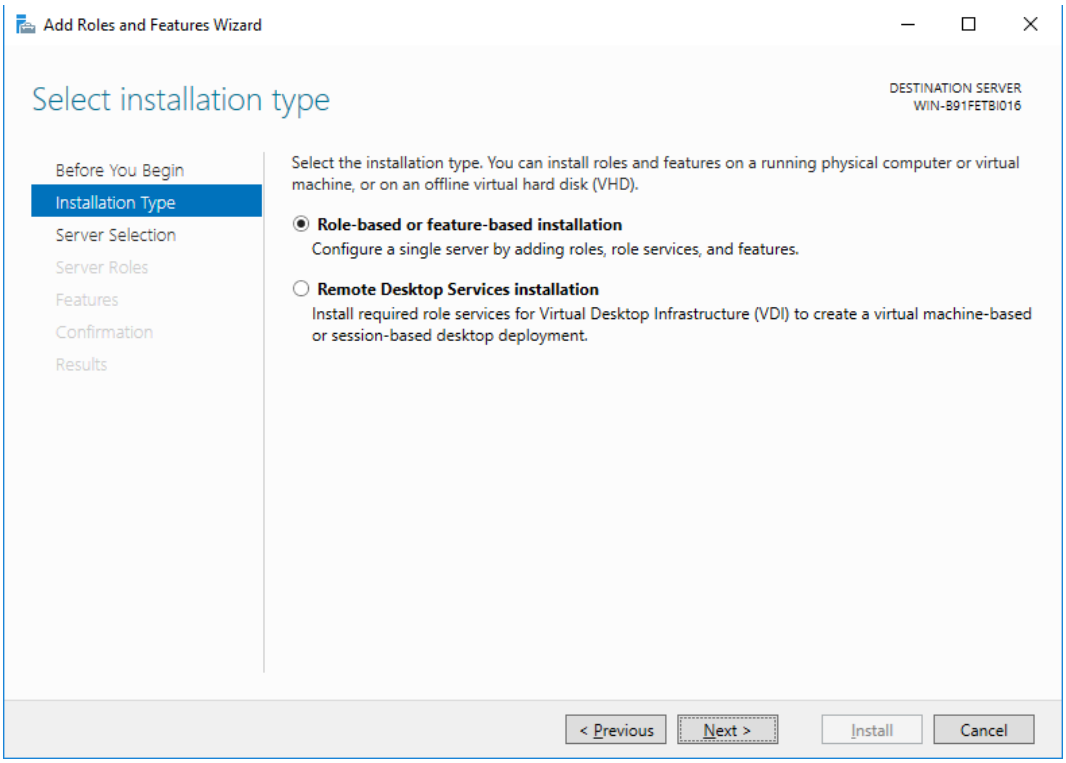
Dhcp szerer beállítása

Nem akar több DHCP-kiszolgálót ugyanazon a hálózaton (úgy véljük, hogy van egy gazember DHCP-szerver a hálózaton), ezért győződjön meg róla, hogy letiltja a szerepét a Virtualbox-ban (feltételezve, hogy Virtualbox, mint én itt vagyok ebben a bemutatóban). Persze, tehetsz olyan fejlett dolgokat, mint például egy biztonsági DHCP szerver, de csak elindulni, ragaszkodni egyet.

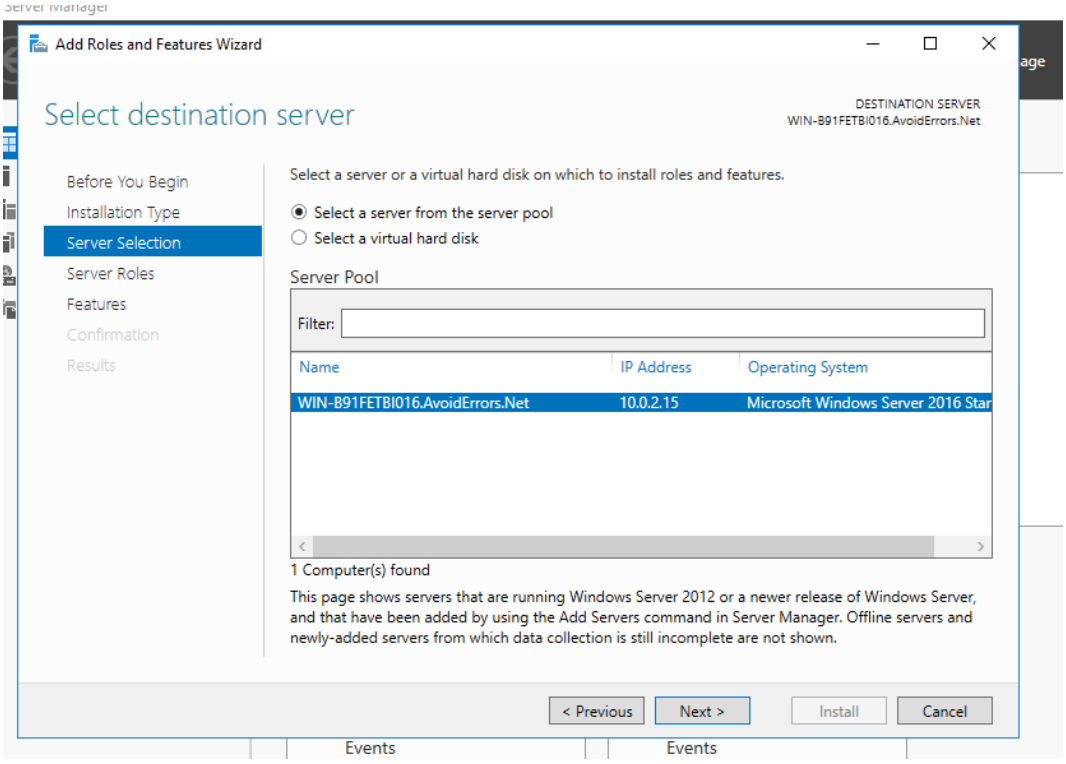
1. A Szerverkezelő irányítópulton kattintson a " Szerepkörök és funkciók hozzáadása"



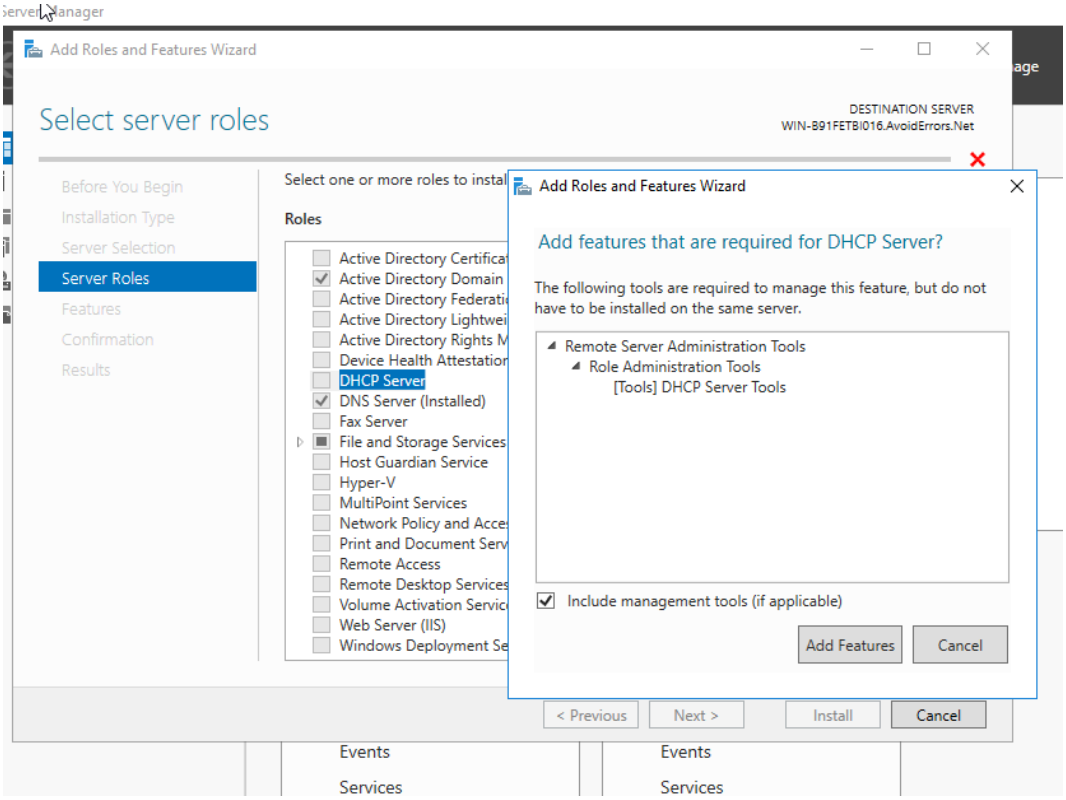
2. Válassza a Szerepkör alapú vagy funkcióalapú telepítést, majd kattintson a Tovább gombra .



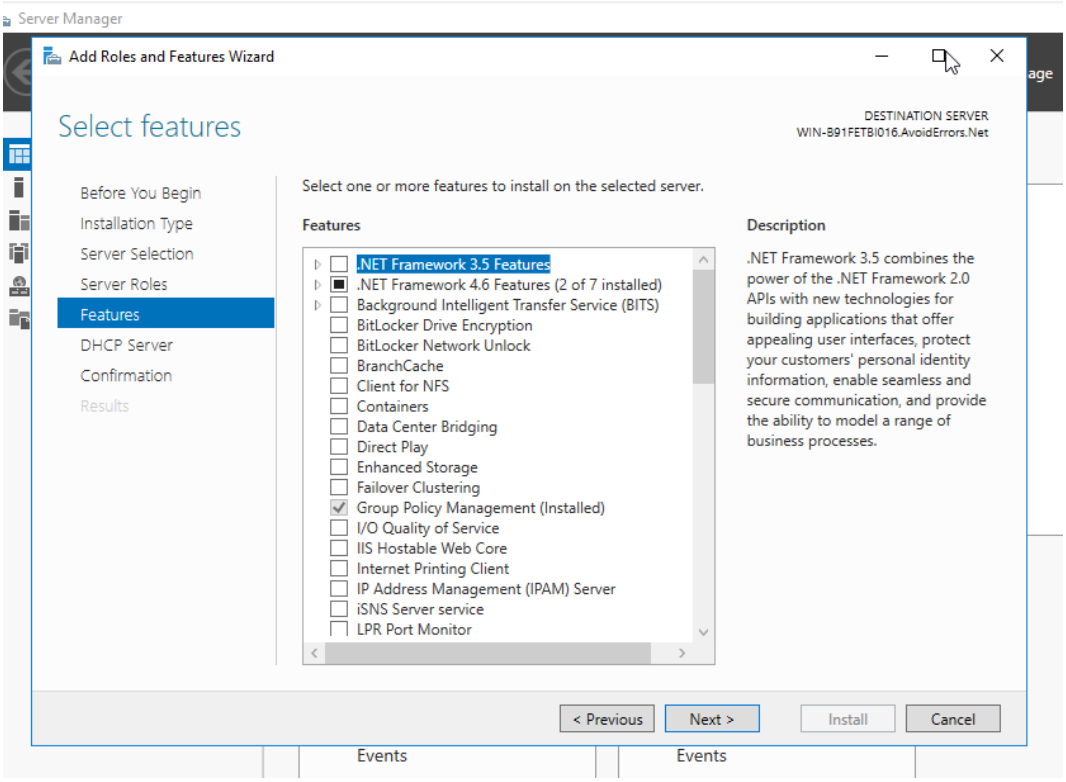
3. Válassza ki a kiszolgálót a "Szerver pool" -ból, majd kattintson a Next gombra .



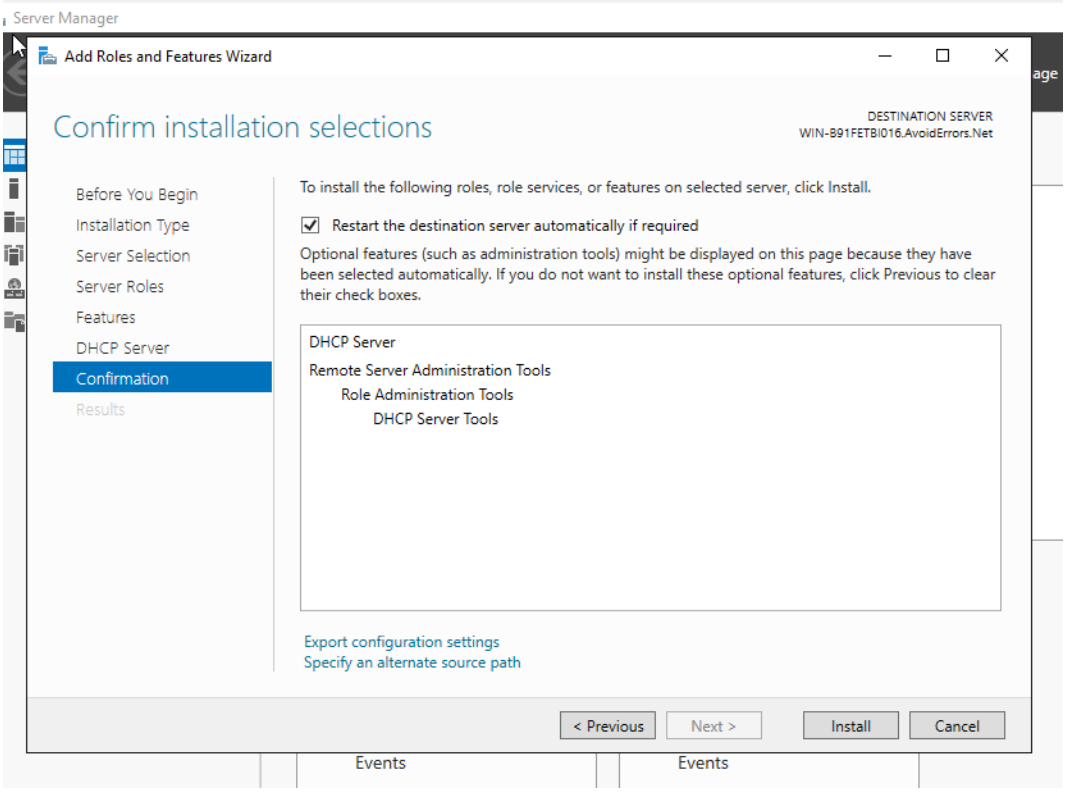
4. ellenőrizze a "DHCP-kiszolgáló" szerepét, és kattintson a "Szolgáltatások hozzáadása" lehetőségre, majd kattintson a "Tovább" gombra.



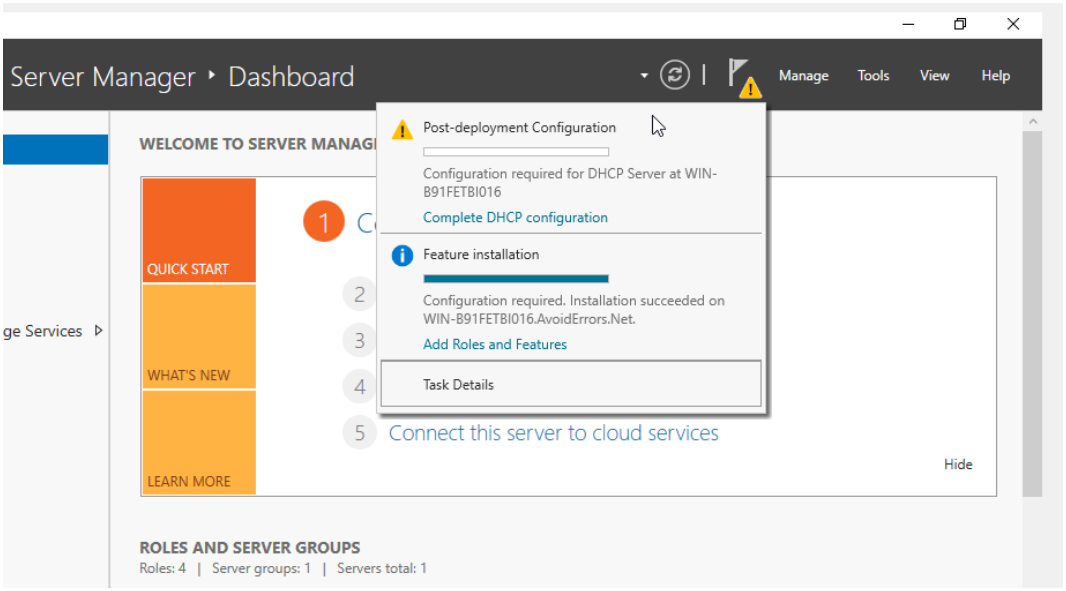
5. A szokásos módon kattintson az "Ezután" gombra



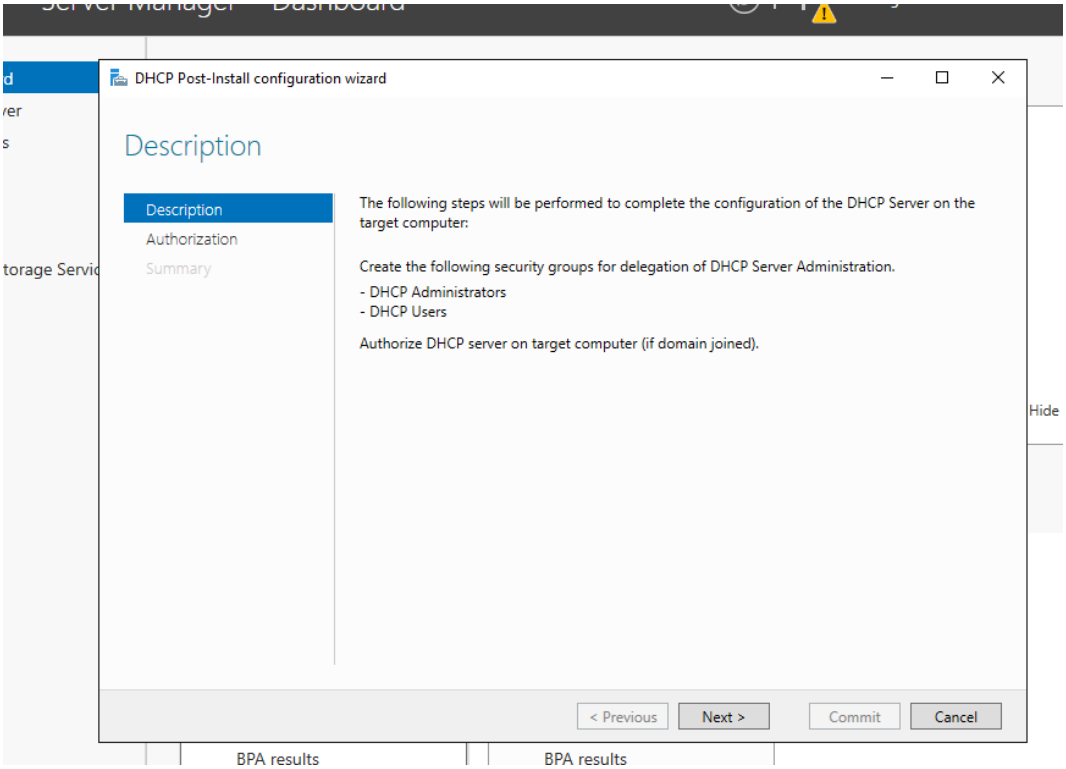
7. Győződjön meg róla, hogy ellenőrizze "Indítsa újra a célkiszolgálót, ha szükséges", majd kattintson a Telepítés gombra.



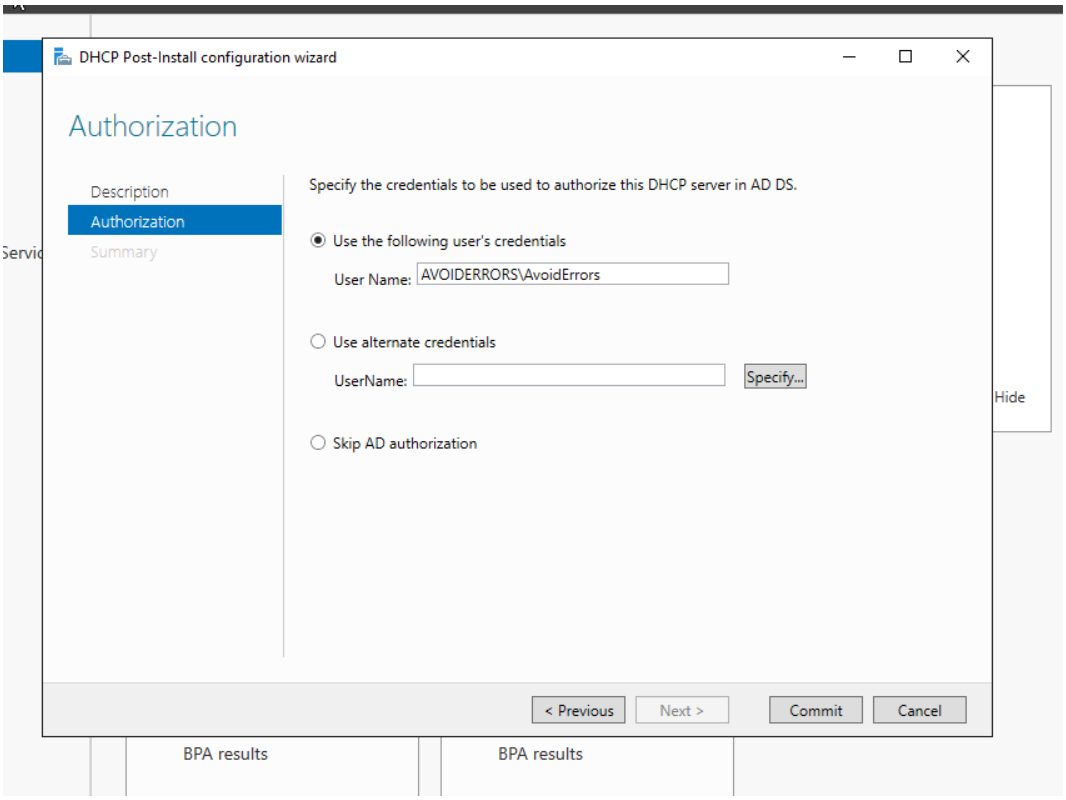
8. a kiszolgáló újraindítása után nyissa meg a kiszolgálókezelőt, és észre fogod venni a "sárga felkiáltójel háromszöget", kattintson rá, majd kattintson a Teljes DHCP beállítás



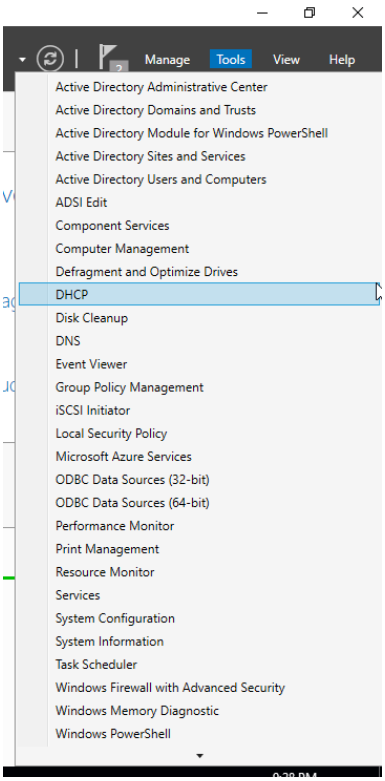
9. ebben a szakaszban kattintson a "Tovább"



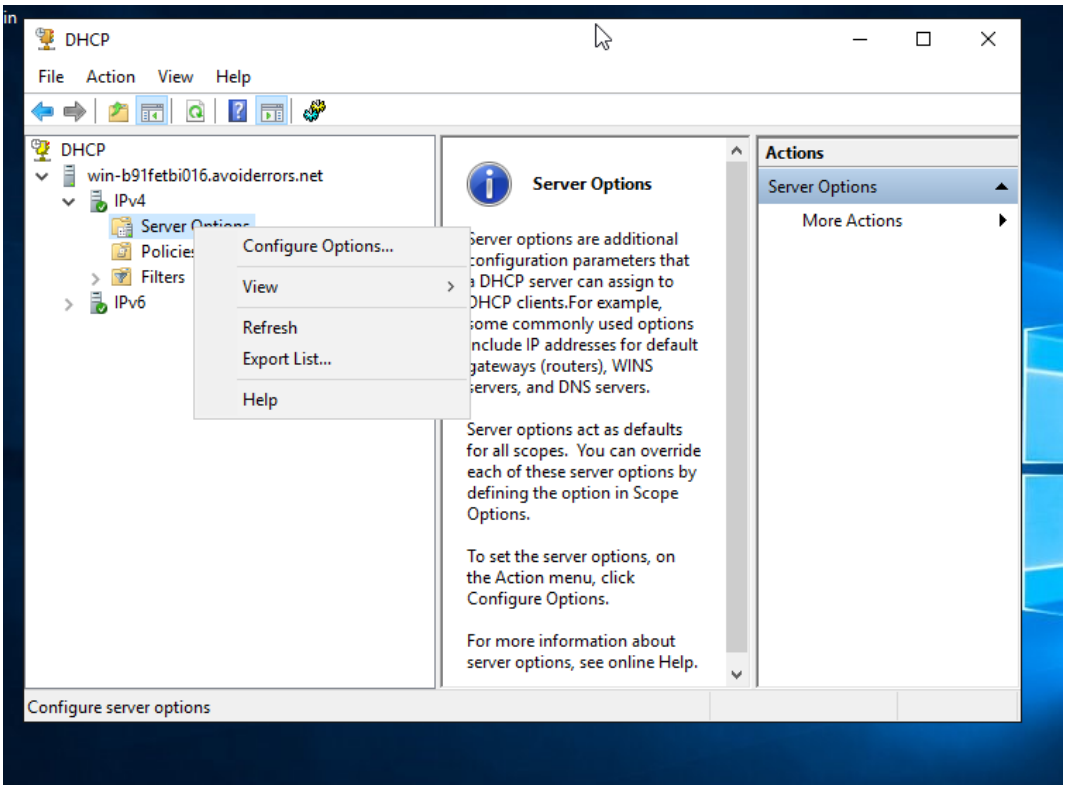
10. ebben a szakaszban kattintson a "Commit"



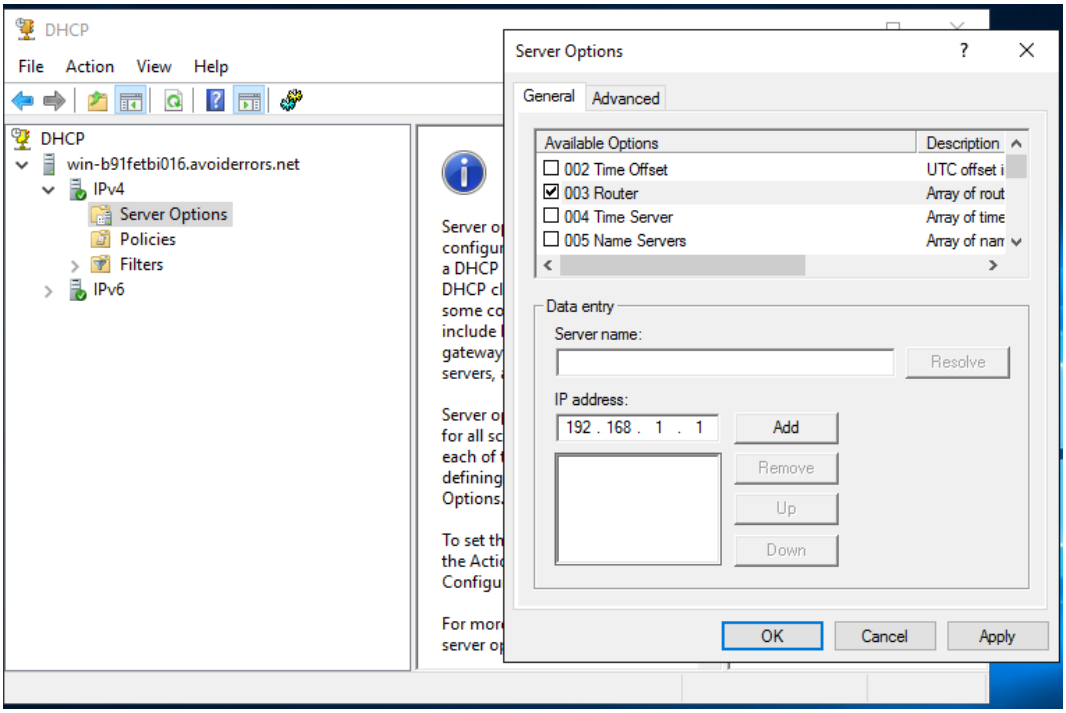
11. A kiszolgálókezelő irányítópultján kattintson a jobb felső sarokban található Eszközök elemre, majd a DHCP gombra a DHCP MMC indításához.



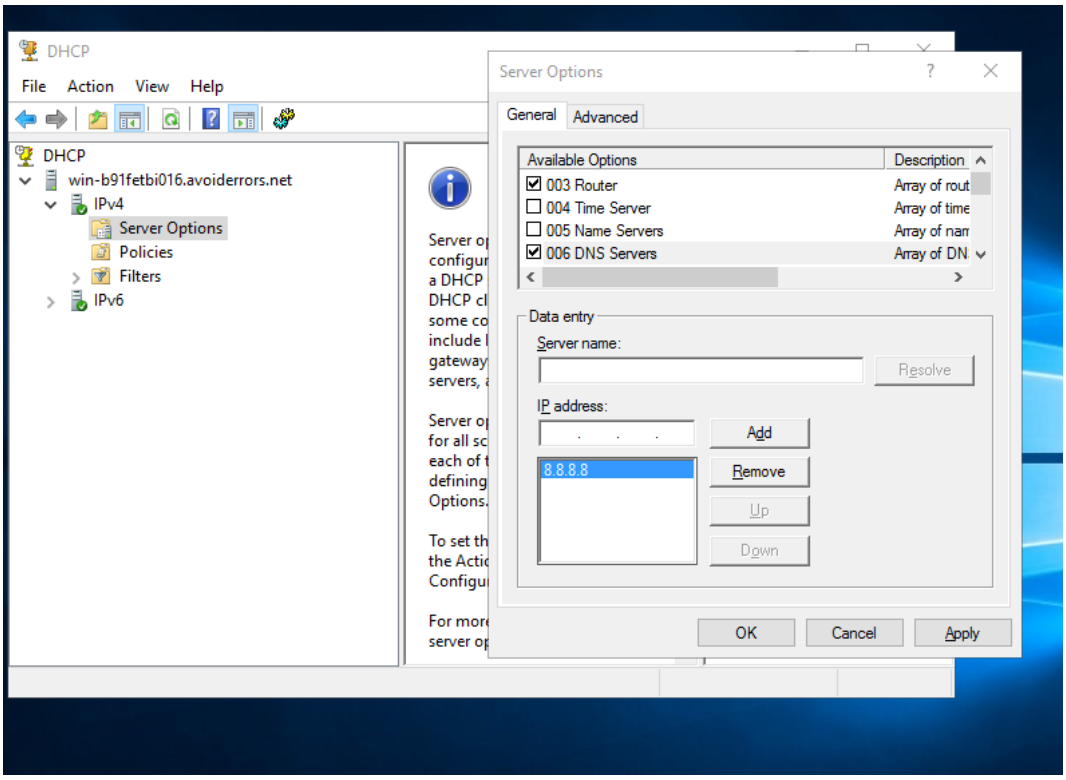
12. Expand DHCP | "Kiszolgáló neve" .Avoiderrors.net | IPv4, jobb egérgombbal kattintson a Kiszolgáló beállításai lehetőségre, és kattintson a Beállítások megadása lehetőségre.



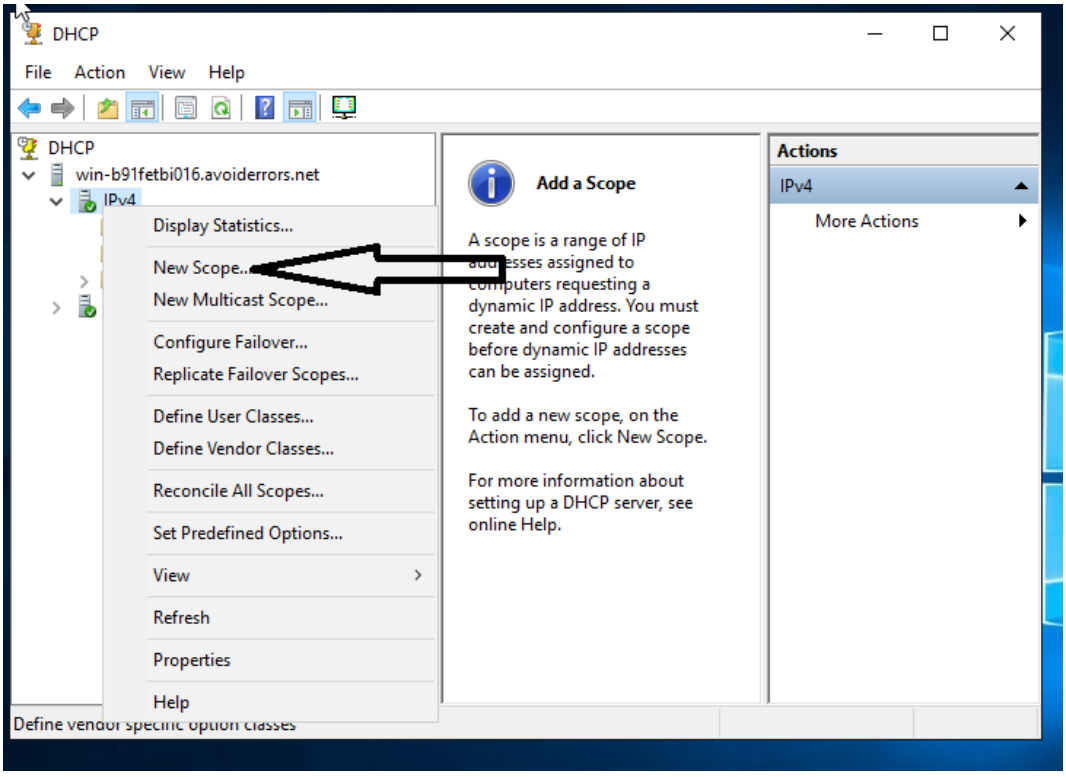
13. ellenőrizze a "003 Router" -t, és alapértelmezés szerint írja be az útválasztó IP-címét a legtöbb "192.168.1.1" routeren, majd kattintson az "Add"



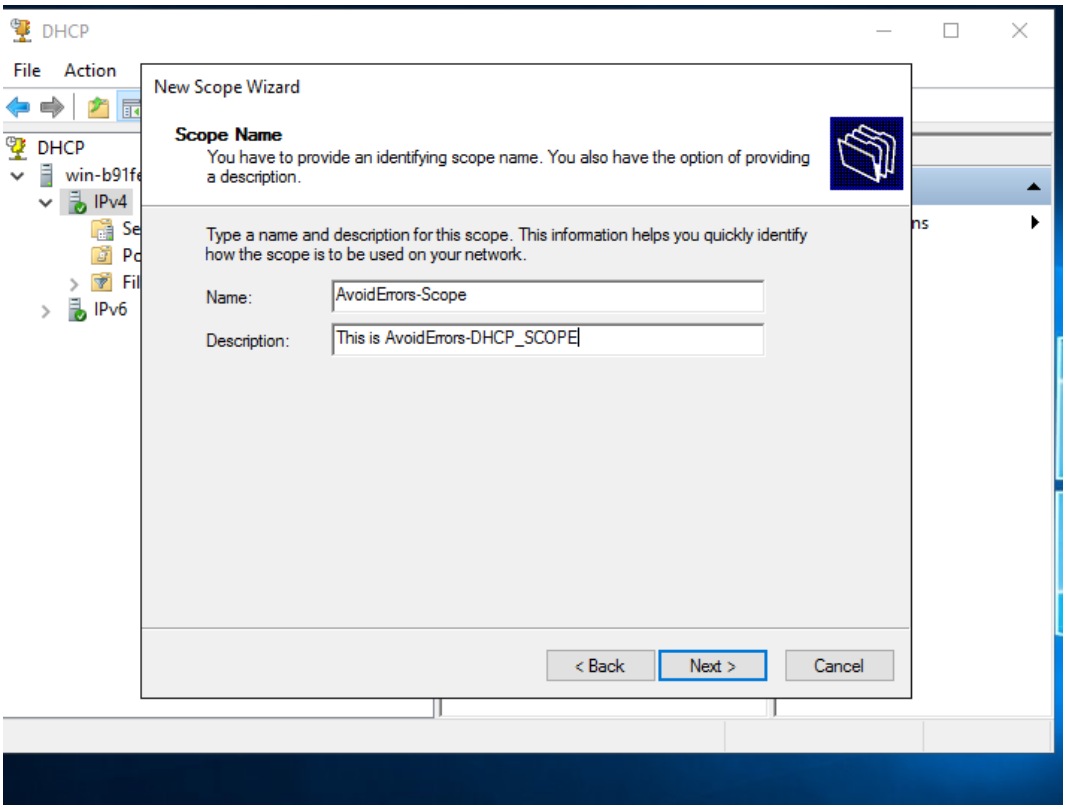
14. Ellenőrizze a "006 DNS-kiszolgálókat", és írja be a kiszolgáló IP-címét az IP cím mezőbe (már létrehoztunk egy DNS-kiszolgálót, amikor telepítettük az AD-szerepünket), számomra írom a google nyilvános DNS-címét "8.8.8.8 ", Majd kattints a Hozzáadás gombra, majd az" OK "



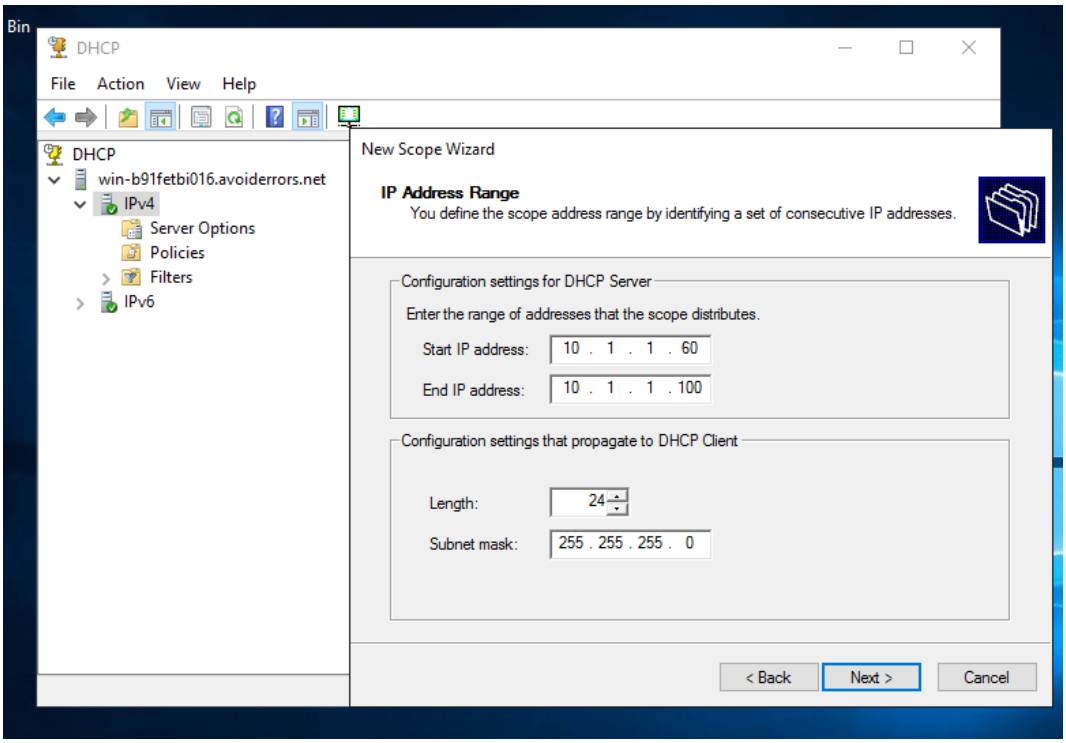
15. most hozzunk létre egy speciális hatókört, jobb klikk az "IPv4" -re, és válasszunk új kört



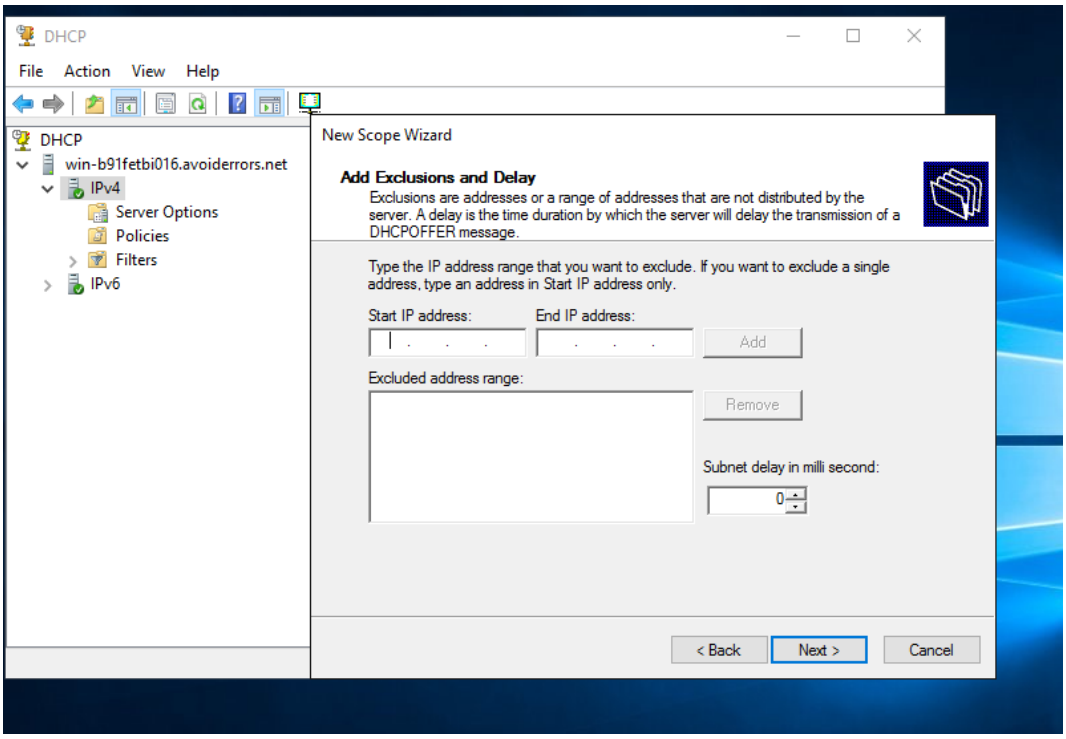
16. miután rákattintott az új hatókörre, majd a következőre, adjon nevet például "AvoidErrors-Scope"



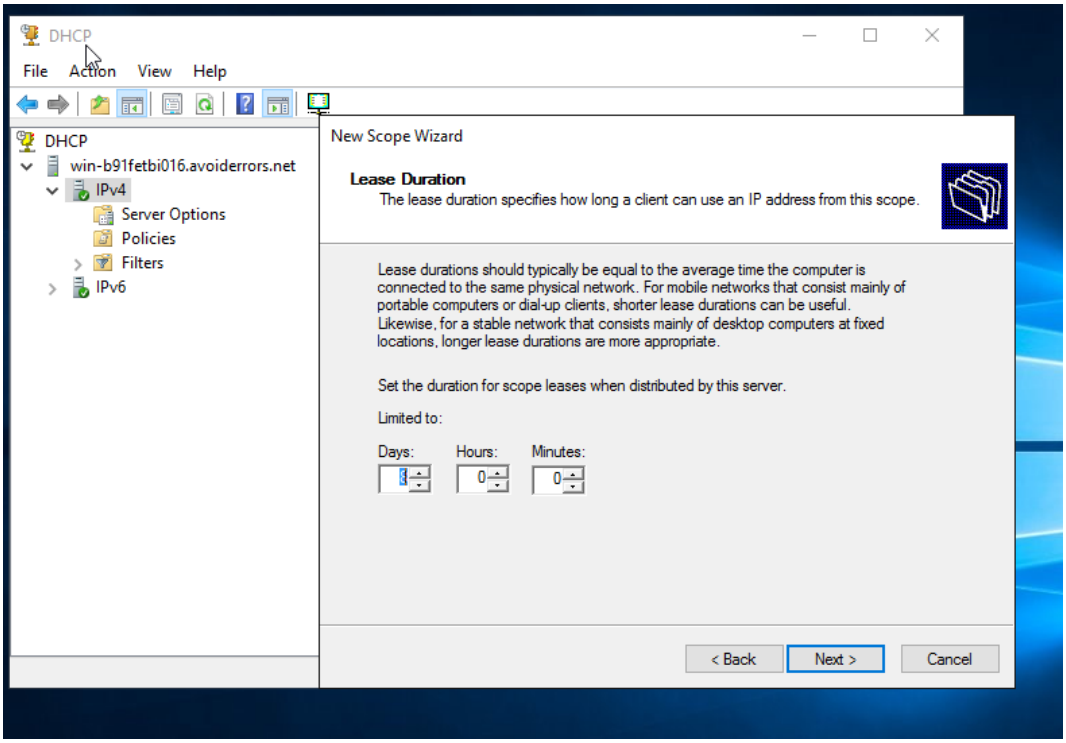
17. Adja meg az IP-címtartományt (indítás és befejezés) és a megfelelő alhálózati maszkot, majd kattintson a Tovább gombra. "Ez az IP címek tartománya, amelyet a szerver átad a DHCP-ügyfeleknek, amelyek a hálózatra érkeznek".



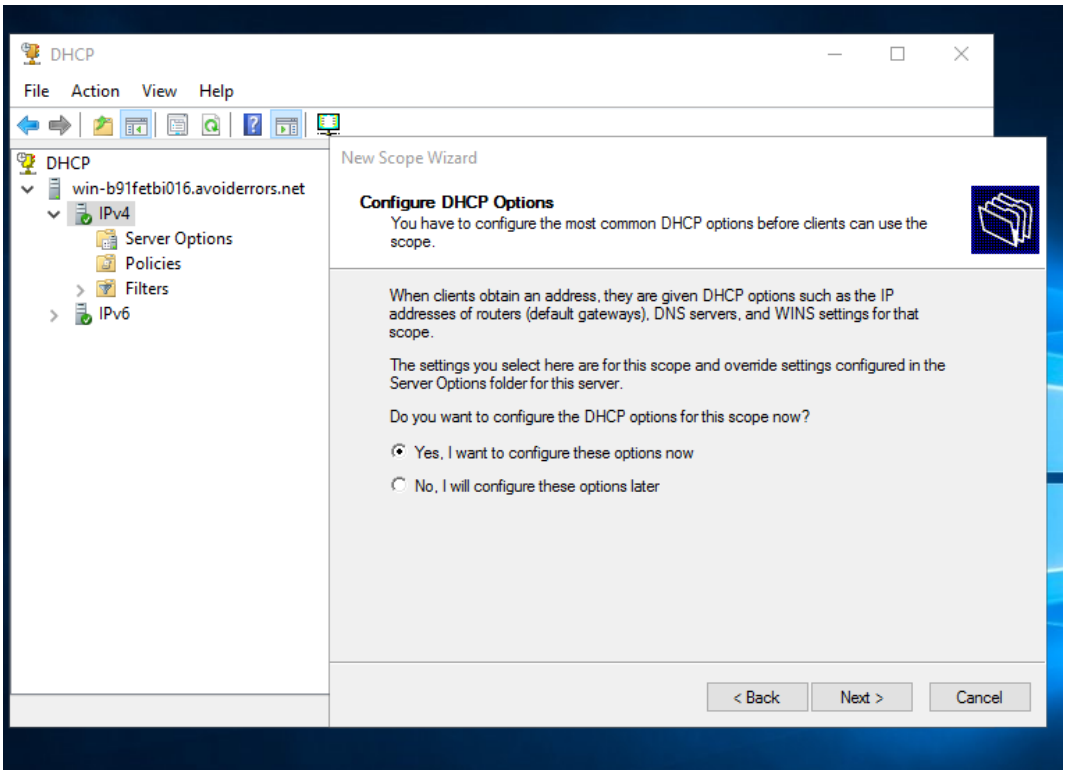
18. A kivételek és a késleltetés üresen hagyása, ha nem tervezi, majd kattintson a Tovább gombra.



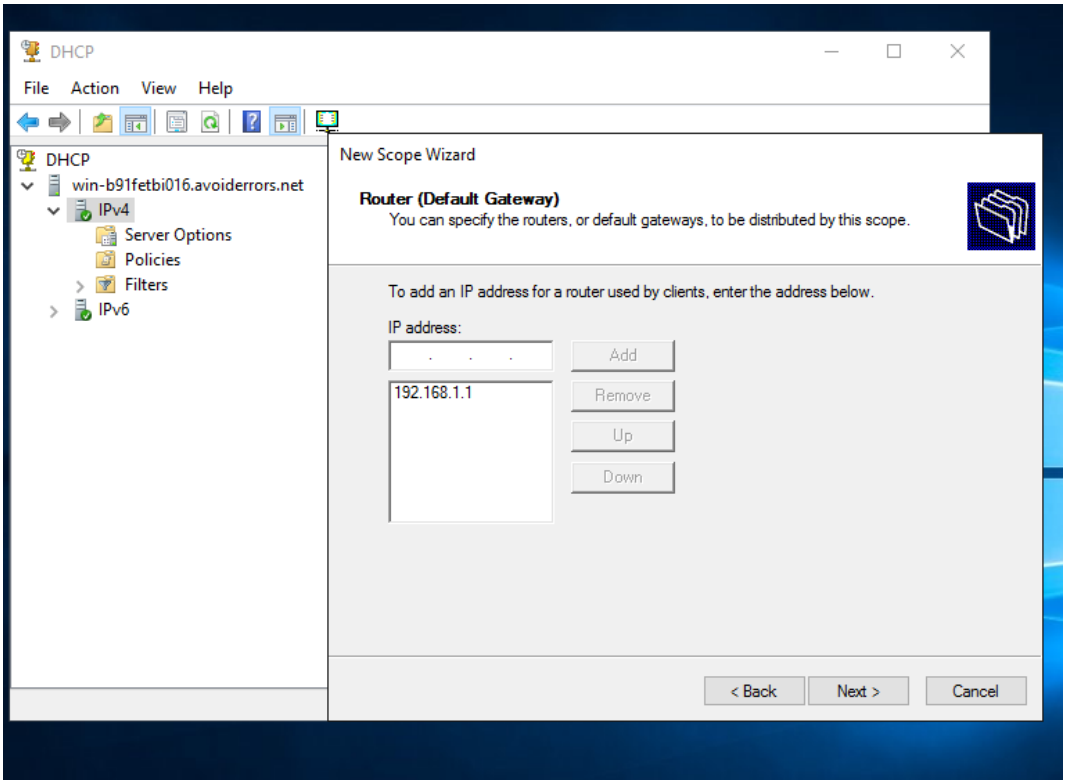
19. Kattintson a Következő a bérleti időtartamra, "hagyja alapértelmezésként, jobb"



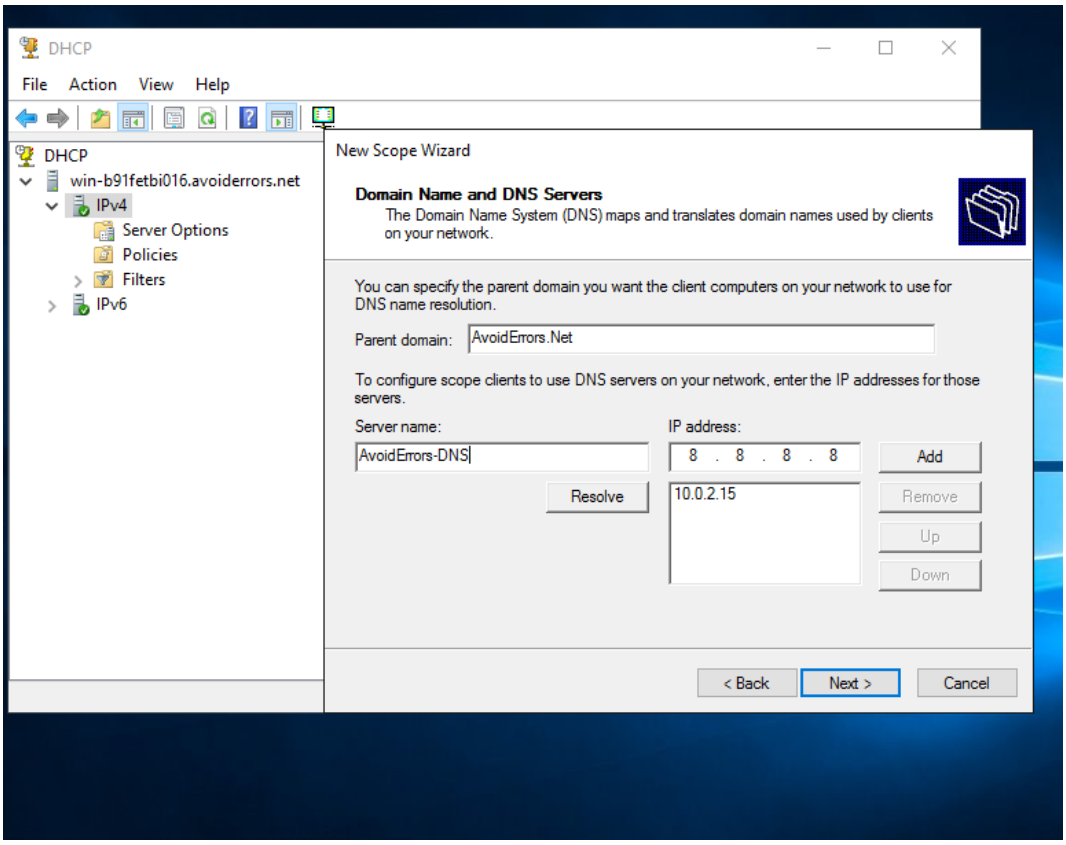
20. Miután rákattintott a "Next" (Következő) gombra, válassza a Yes (Igen) lehetőséget, ezeket a beállításokat most konfigurálni szeretné, majd kattintson a Next (Tovább) gombra



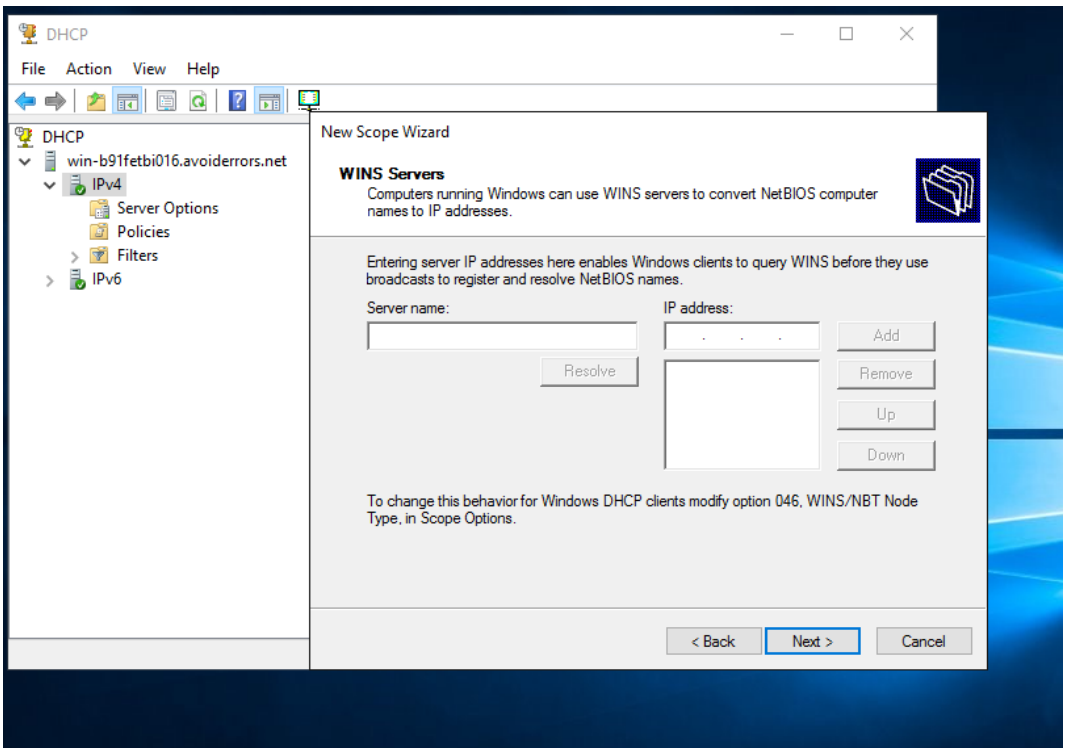
21. a varázsló ezen szakaszában írja be újra a router IP-címét (ne feledje, ez ugyanaz, mint a szerver jelenlegi átjáró címe), majd kattintson a Tovább gombra.



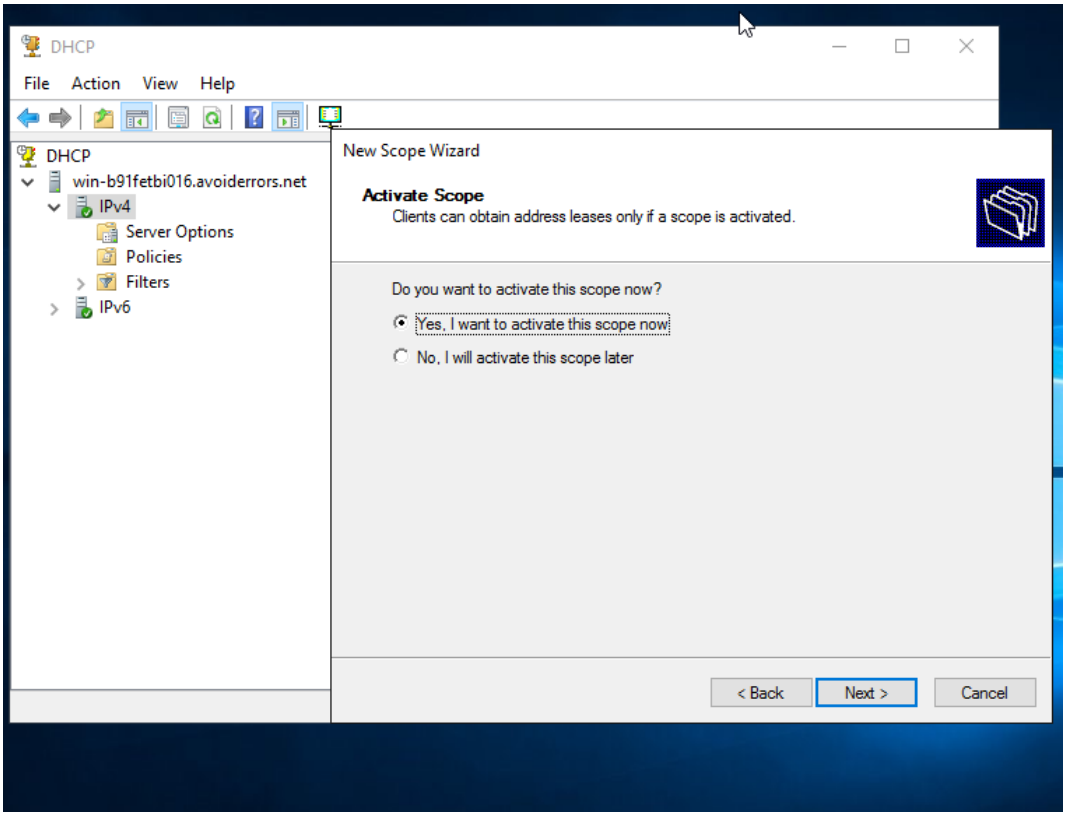
22. És a DNS-kiszolgáló IP-címét már be kell tölteni (google DNS-jünk), kattintsunk a Tovább gombra



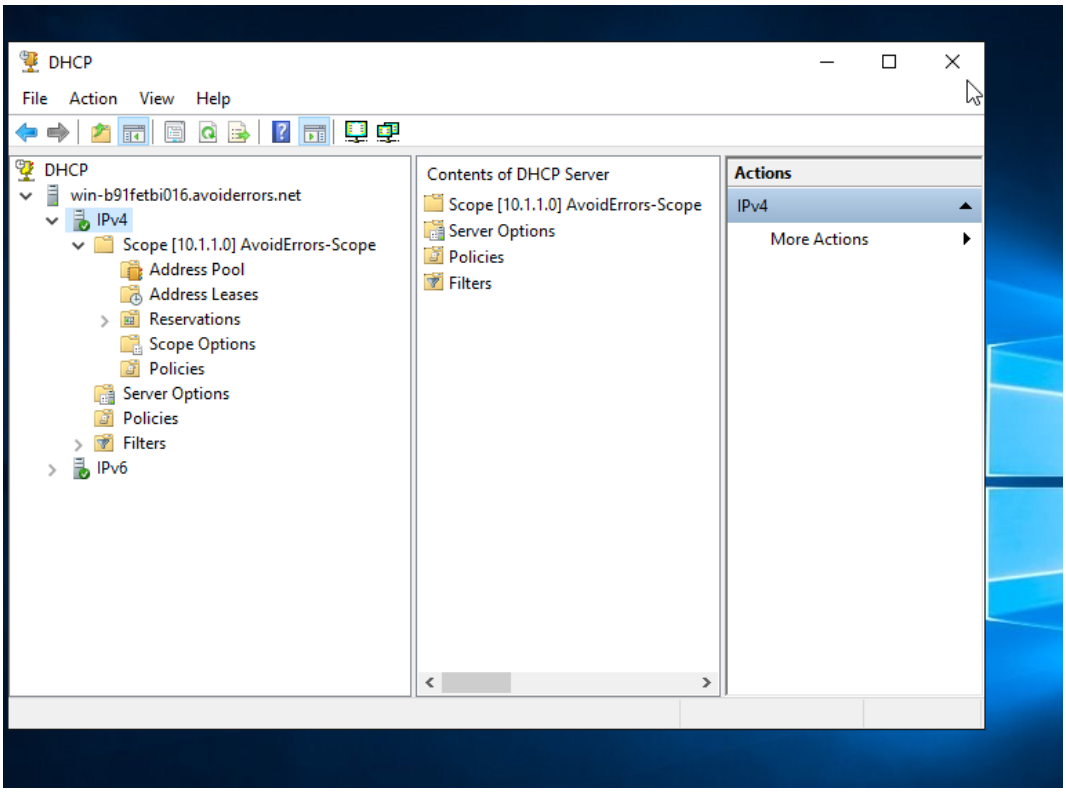
23. hagyja el a varázsló ezen szakaszát, úgy, hogy (üres), mivel nem tervezzük a WINS használatát, kattintsunk a Tovább gombra.



24. A varázsló megkérdezi, hogy ezt az opciót most vagy később szeretné-e aktiválni. Válassza az Igen lehetőséget, most be szeretné kapcsolni ezt az opciót, majd kattintson a Tovább gombra, és a Befejezés



25. Most a köre beállítva és aktiválva van



Ez az! A DHCP az új kiszolgálón van beállítva, és készen áll a használatra. Ismét meg kell győződnie arról, hogy letiltja a DHCP-kiszolgálóval azonos hálózaton működő többi DHCP-kiszolgálót. Nem kíván több DHCP-kiszolgálót hálózaton.

DFS – Elosztott fáljrendszer

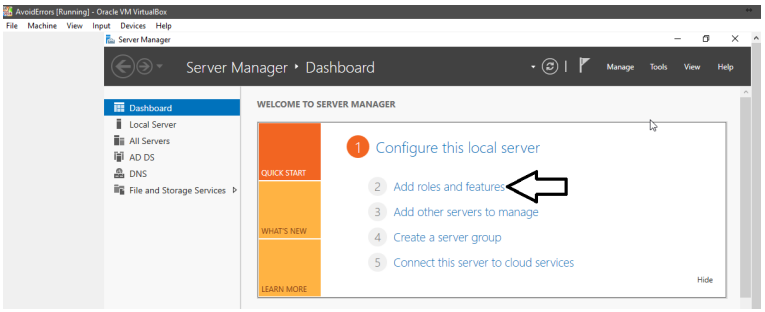
Az elosztott fájlrendszer (DFS) olyan ügyfél- és kiszolgálószolgáltatáskészlet, amely lehetővé teszi a Microsoft Windows szervereket használó szervezet számára, hogy számos elosztott SMB fájlmegosztást rendezzen el egy elosztott fájlrendszerbe.

A DFS lehetővé teszi a helyek átláthatóságát (a névtér-összetevőn keresztül) és a redundanciát (a fájl-replikációs összetevőn keresztül) annak érdekében, hogy javuljon az adatok rendelkezésre állása a hiba vagy a nagy terhelés miatt, ha több különböző helyen lévő megosztásokat logikusan csoportosíthatja egy mappában vagy DFS-gyökérben.

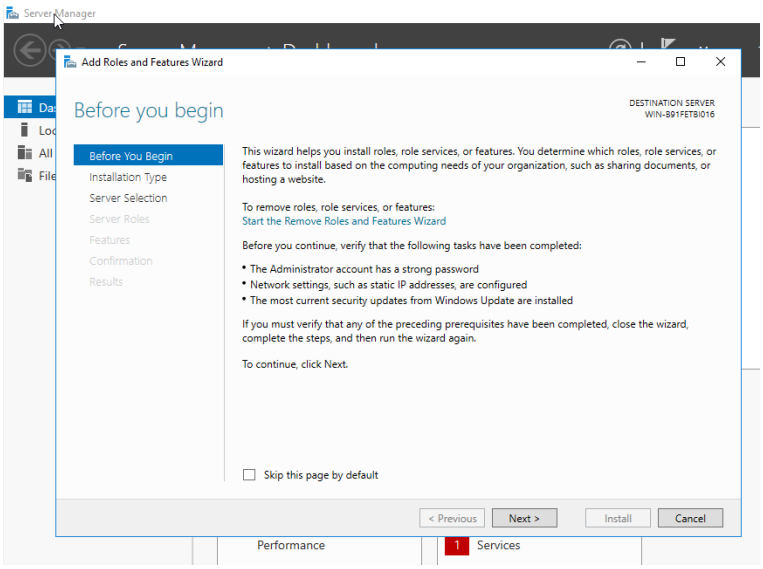
A Microsoft DFS-jét a Microsoft "DFS" és "Dfs" kifejezések helyettesítik, és nincs kapcsolatban a DCE elosztott fájlrendszerrel, amely a DFS védjegyet tartotta, de 2005-ben megszűnt.

Ebben a bemutatóban telepítjük és konfiguráljuk a DFS (Distributed File System) névtereket a Windows Server 2016-ban. A DFS lehetővé teszi, hogy a különböző kiszolgálókon tárolt megosztott mappákat egy vagy több logikailag strukturált névteret hozzon létre. Minden névtér a felhasználók számára egy megosztott mappaként jelenik meg, és egy sor almappával rendelkezik. Ez a beállítás növeli a rendelkezésre állást, és automatikusan összekapcsolja a felhasználókat a megosztott mappákkal ugyanazon az Active Directory Domain Services webhelyen.

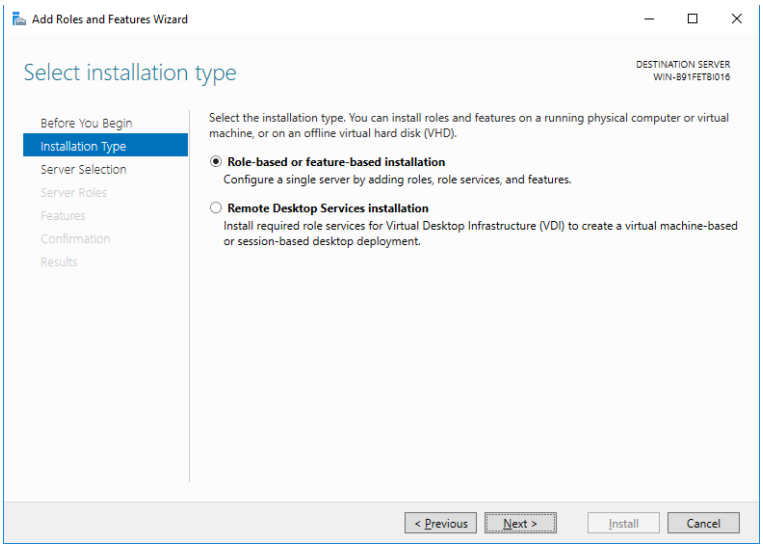
1. lépés: Jelentkezzen be a Windows szerverre és nyissa meg a "Server Manager" parancsot, majd nyomja meg a "Róluk és funkciók hozzáadása"



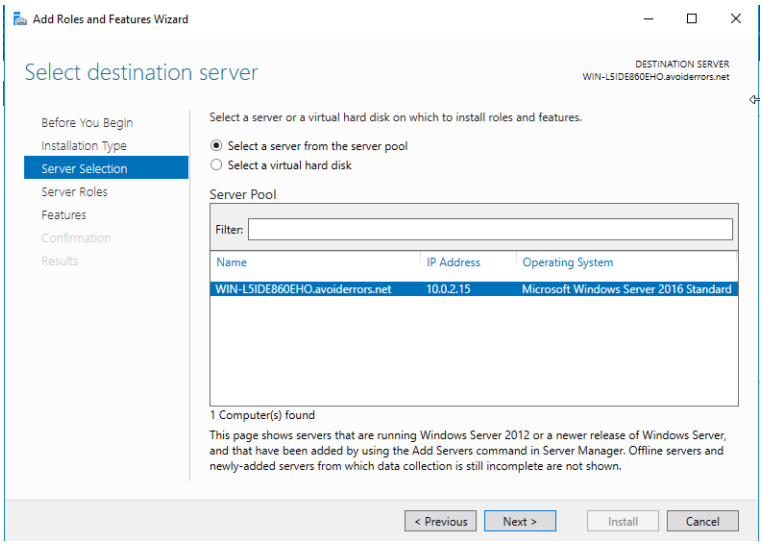
A szokásos módon az "Add szerepek" varázsló megjelenik az alábbiak szerint



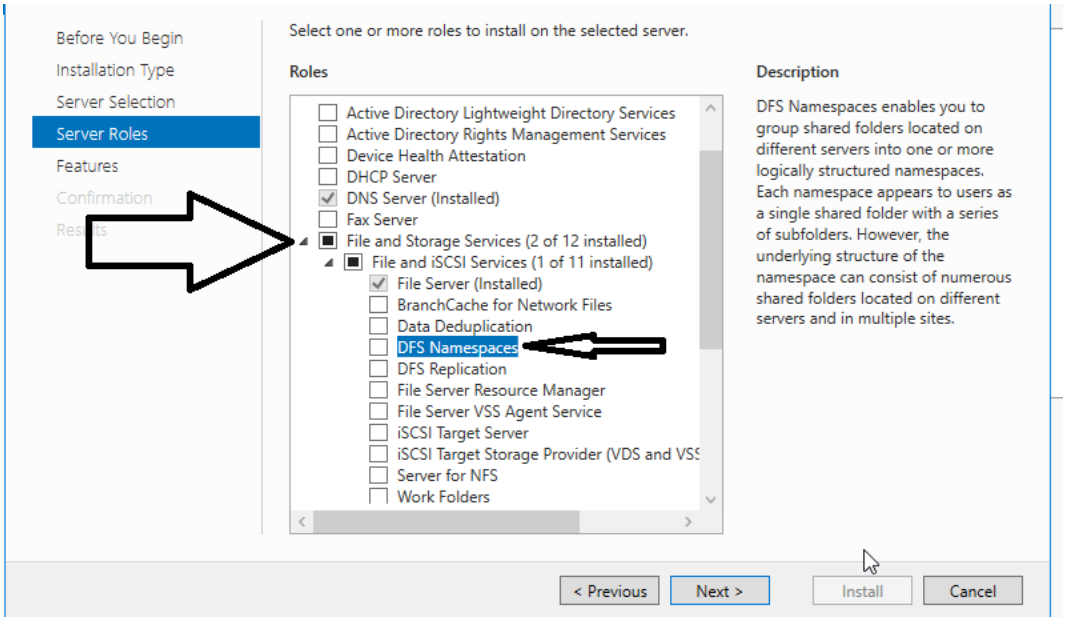
Kattintson a "KÖVETKEZŐ" elemre, és válassza a "Szerepkör alapú vagy funkcióalapú telepítés"



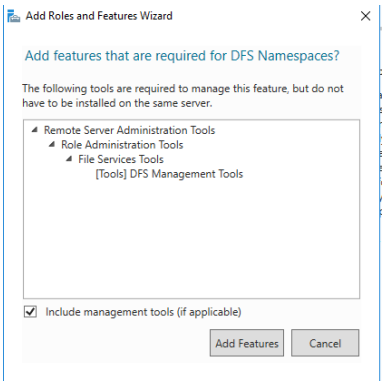
Nyomja meg a "Tovább" gombot, majd válassza ki a kiszolgálót, majd kattintson ismét a "Tovább" gombra.



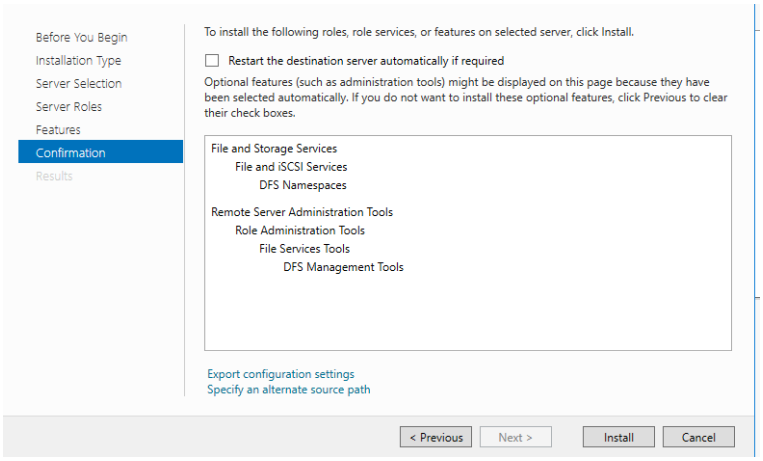
A listából válassza ki a " Fájl és tárolási szolgáltatások " pontot, és válassza ki a listából az " DFS névterek " lehetőséget .



Miután kiválasztottad, az alábbi módon fog megjelenni, majd kattintson a "Funkciók hozzáadása" gombra.

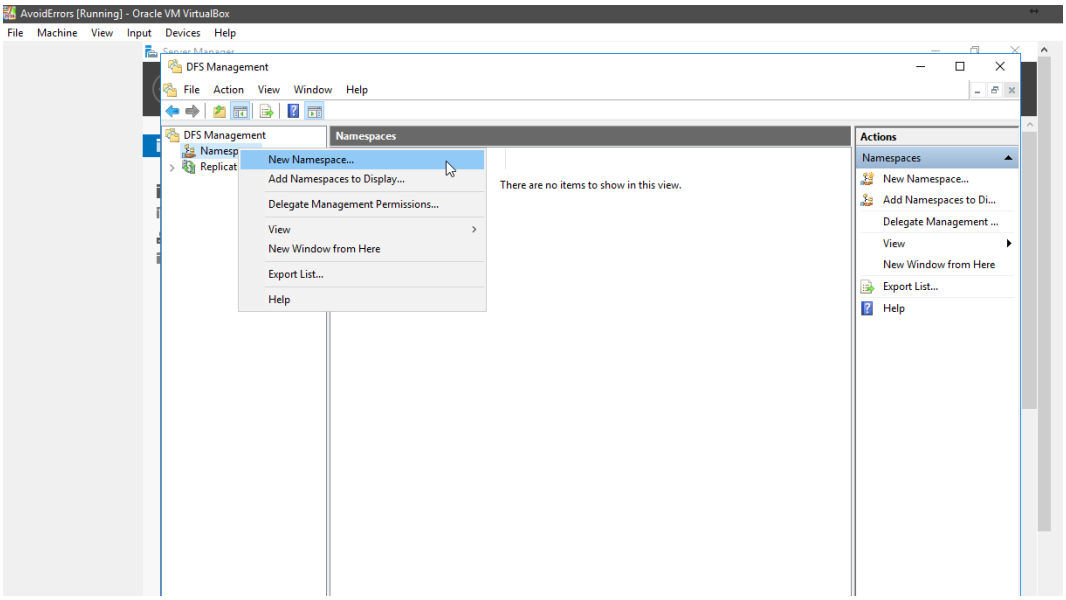


Miután rákattintasz a "Szolgáltatások hozzáadása" gombra, kattintson a "Tovább" gombra a képernyő többi részén, majd nyomja meg a "Telepítés" gombot .

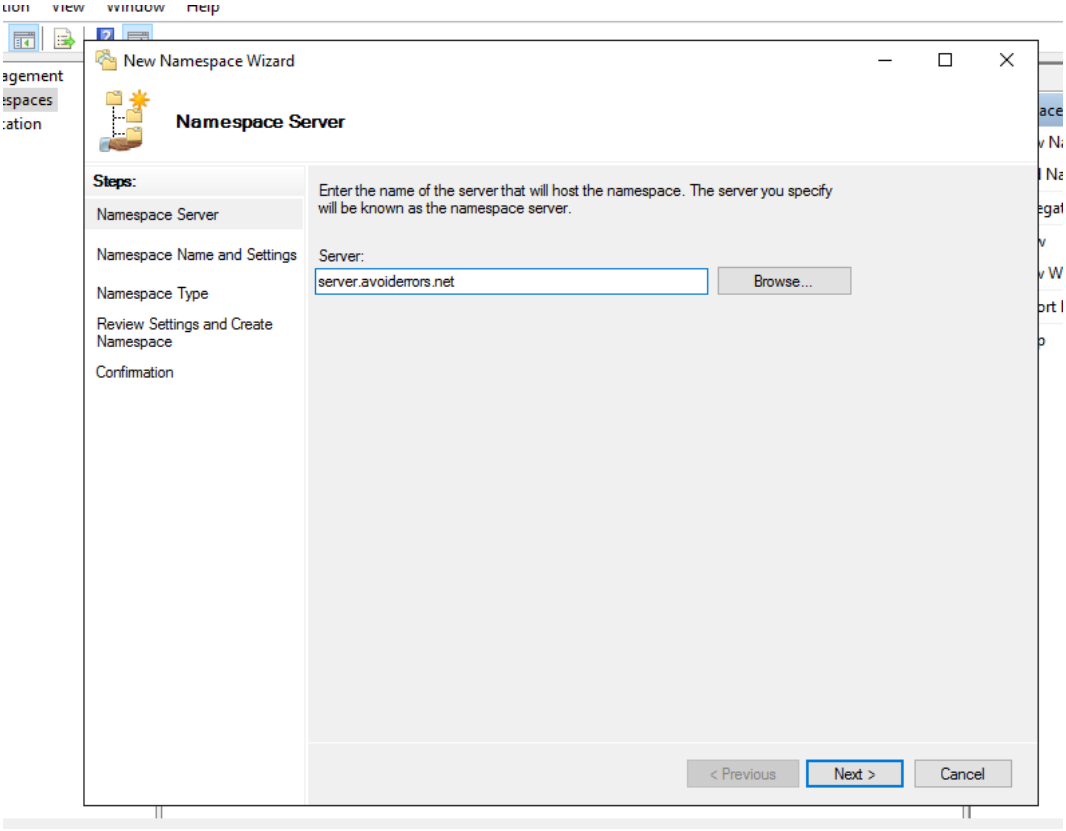


2. lépés: A DFS konfigurálása

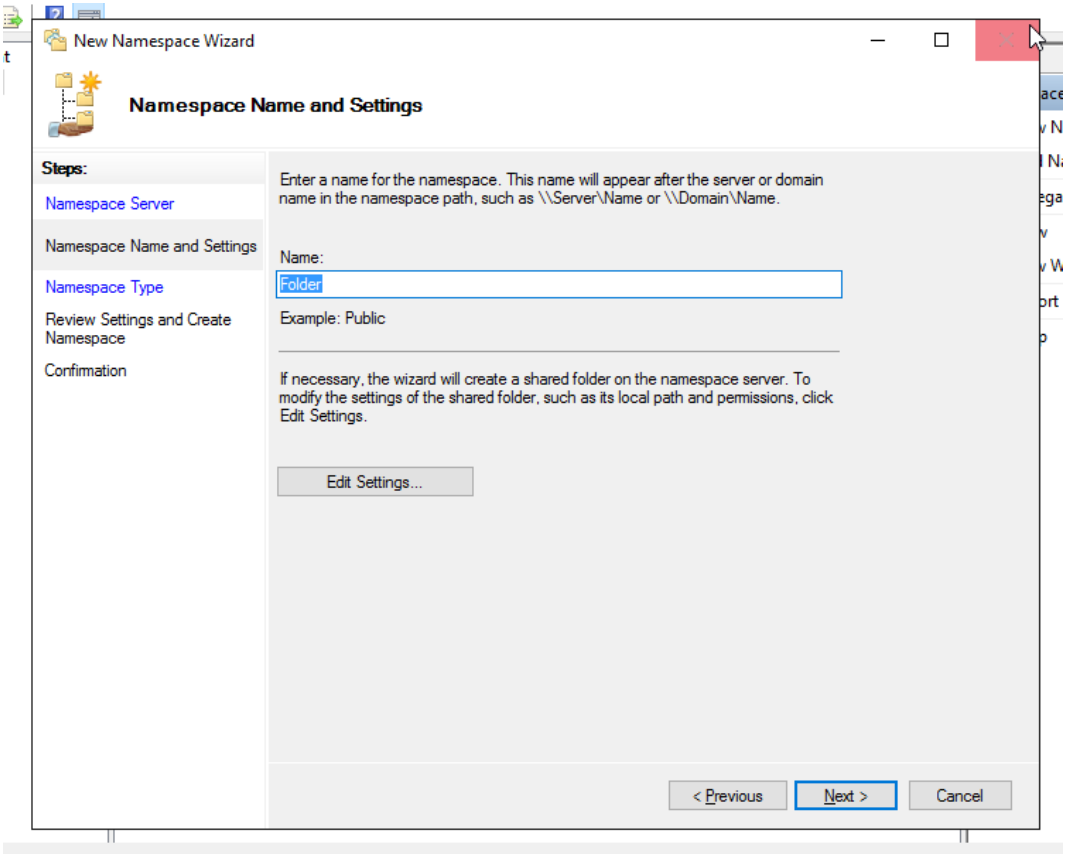
Miután telepítettük a DFS-szerepünket, áttérünk a második lépésre, és a DFS-ről való konfigurálással foglalkozunk a kiszolgálókezelőn az "Eszközök" lehetőségre, és a legördülő listából válassza a " DFS-menedzsment" lehetőséget , majd kattintson a jobb gombbal a Névfüzetek elemre és válassza ki a Új névtér.



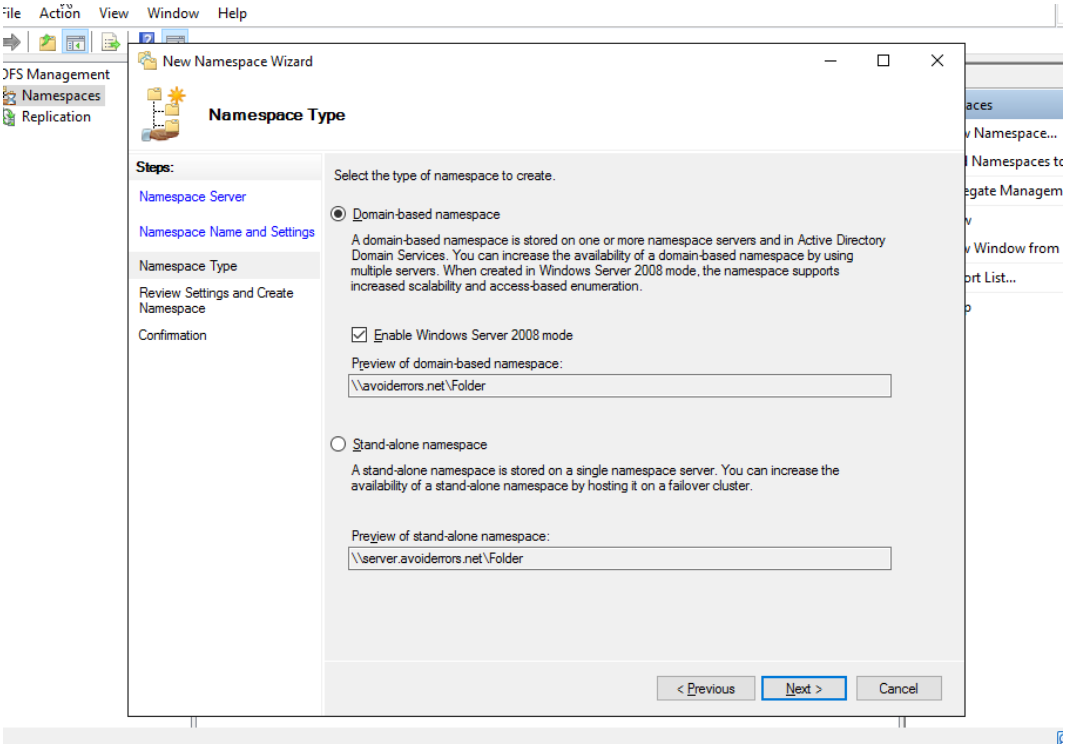
Ezután írja be a szerver nevét, amely a következő képernyőn fogja tárolni a névteret



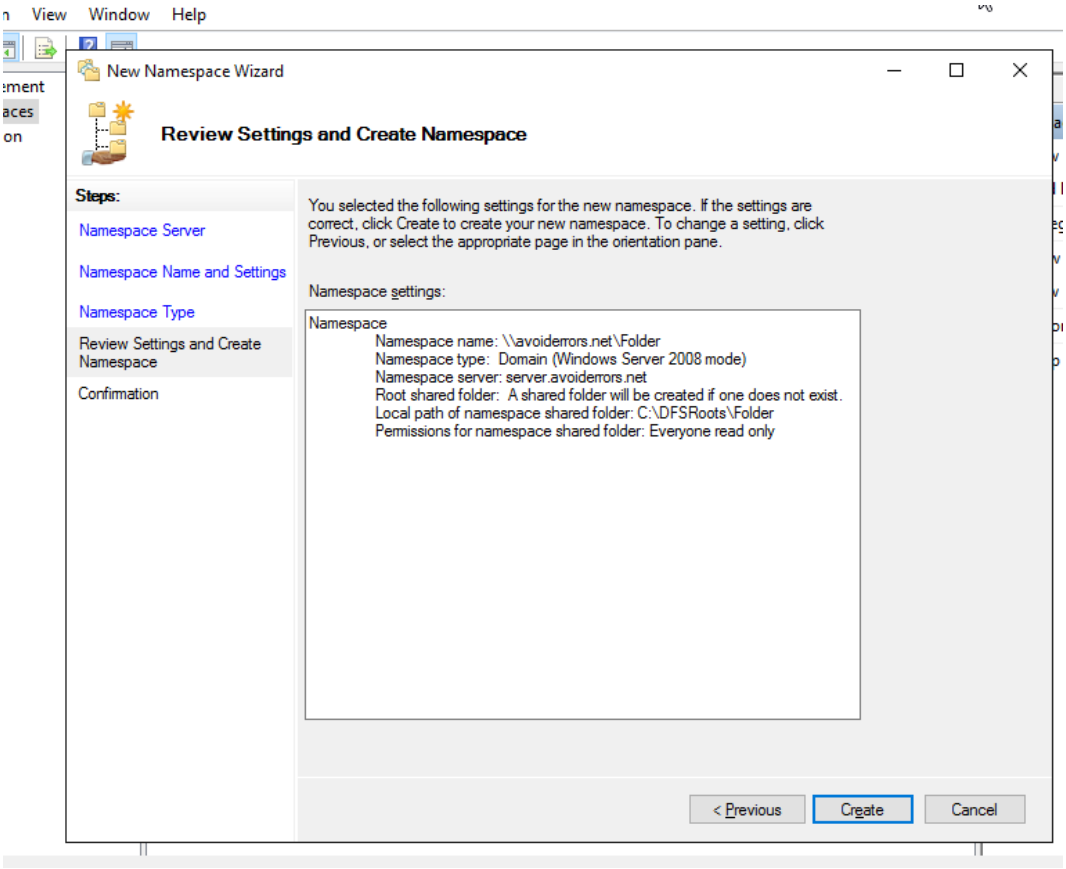
Kattintson a Tovább gombra . Válasszon nevet a névtérhez a következő képernyőn. ez lesz a domain megosztási útvonalának neve. például avoiderrors.net/folder, kattintson a Beállítások szerkesztése gombra a megosztás engedélyeinek módosításához. alapértelmezés szerint mindenkinek csak "olvasott" engedélyei vannak. kattintson a Tovább gombra . a következő képernyőn válassza ki a Névtér típusát.



Válassza ki a Domain alapú névteret, majd kattintson a Tovább gombra .



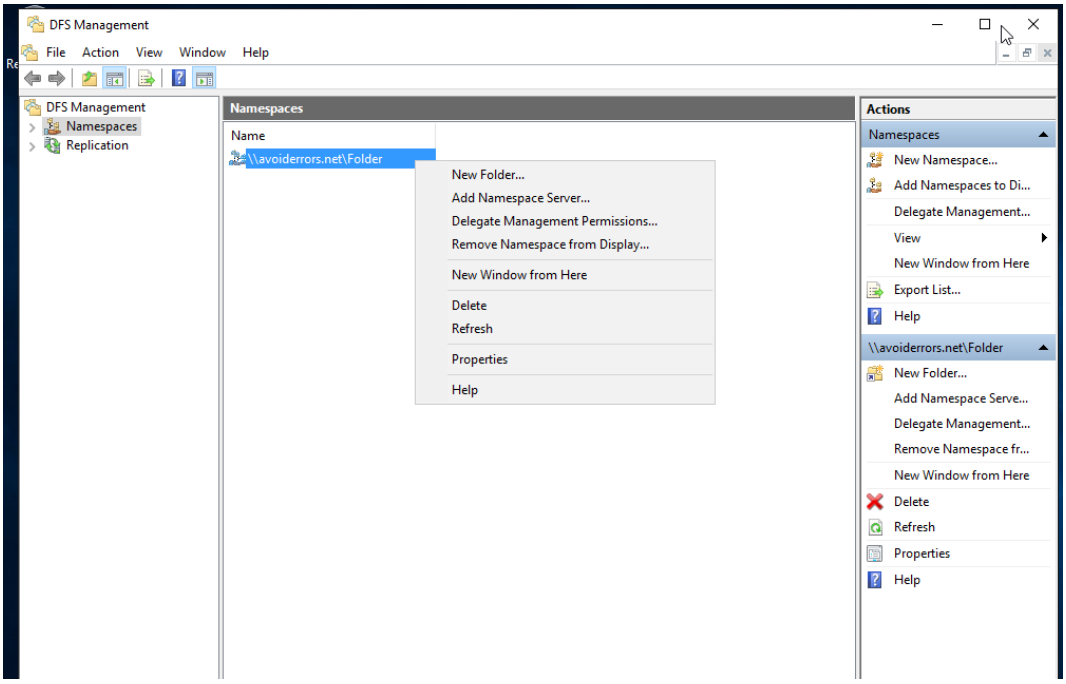
Tekintse át a beállításokat, majd kattintson a Létrehozás gombra :



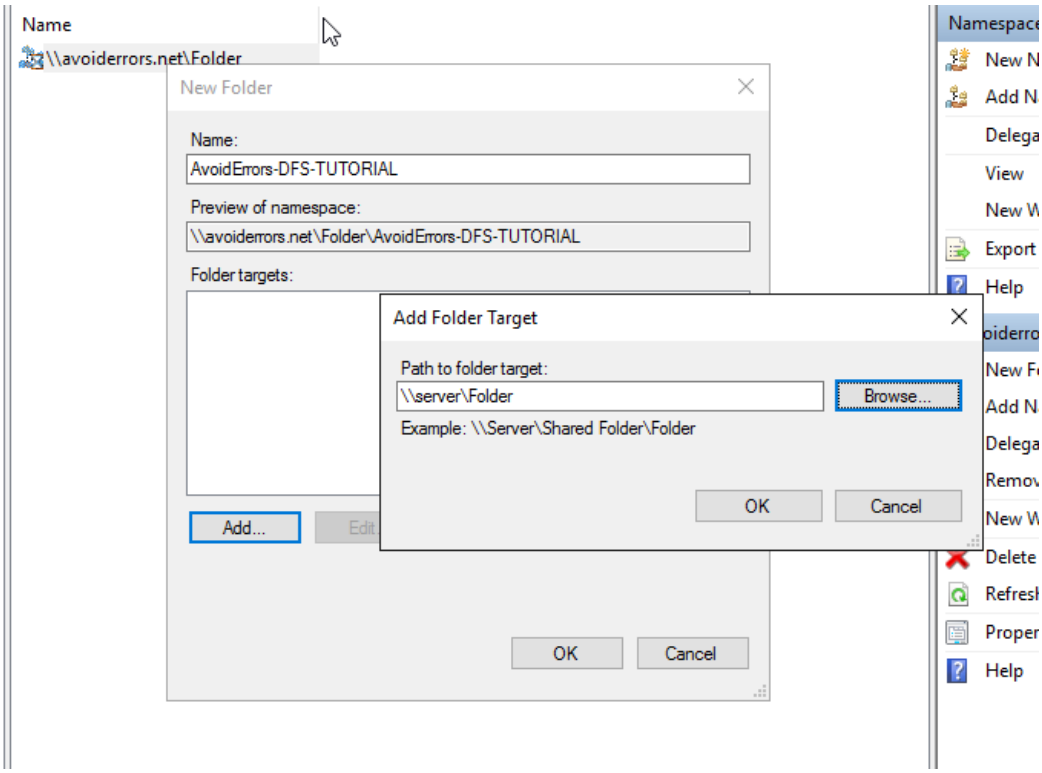
Tekintse át a beállításokat, majd kattintson a Létrehozás gombra .

3. lépés: DFS-mappa létrehozása

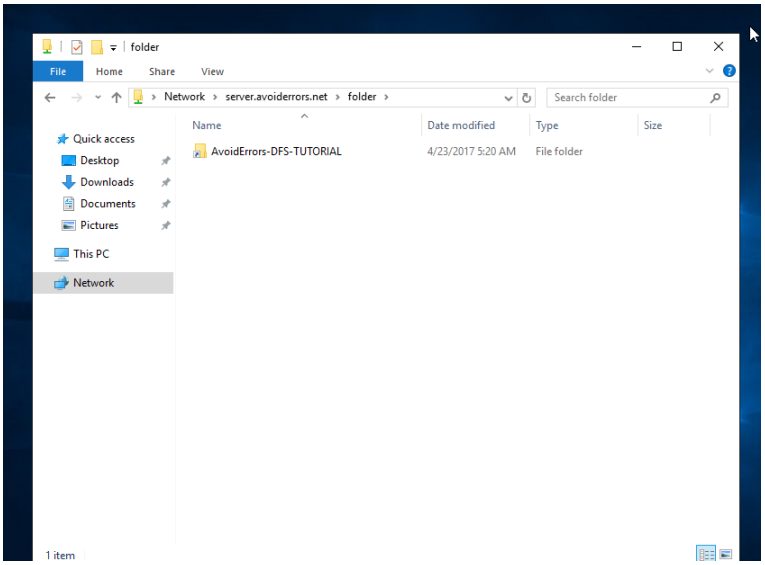
Most hozzáadunk mappákat a DFS névtérhez. a DFS kezelőkonzoltól jobb gombbal kattintson a létrehozott névtérre, és válassza az Új mappa lehetőséget :



Írja be a mappa nevét, majd kattintson a Hozzáadás gombra , és írja be a Névtárhoz hozzáadni kívánt Megosztott mappa elérési útját:



Kattintson az OK gombra, menjünk a hálózati elérési útra (pl. \\ server.avoiderrors.net \ mappa), és látnod kell a hozzátartozó mappát.



Következtetés

A DFS névterek egy nagyszerű lehetőség a Windows szerverben a hálózati megosztások megszervezéséhez. A DFS-névterek használatakor nem számít, hogy hol vannak a megosztott mappák, mindegyik elérhetõ egyetlen elérési útból. Ez megkönnyíti a fájlszerverek mozgatását a hozzáférési utak megszakítása nélkül.

Több IP-cím beállítása a egy hálózati kártyához

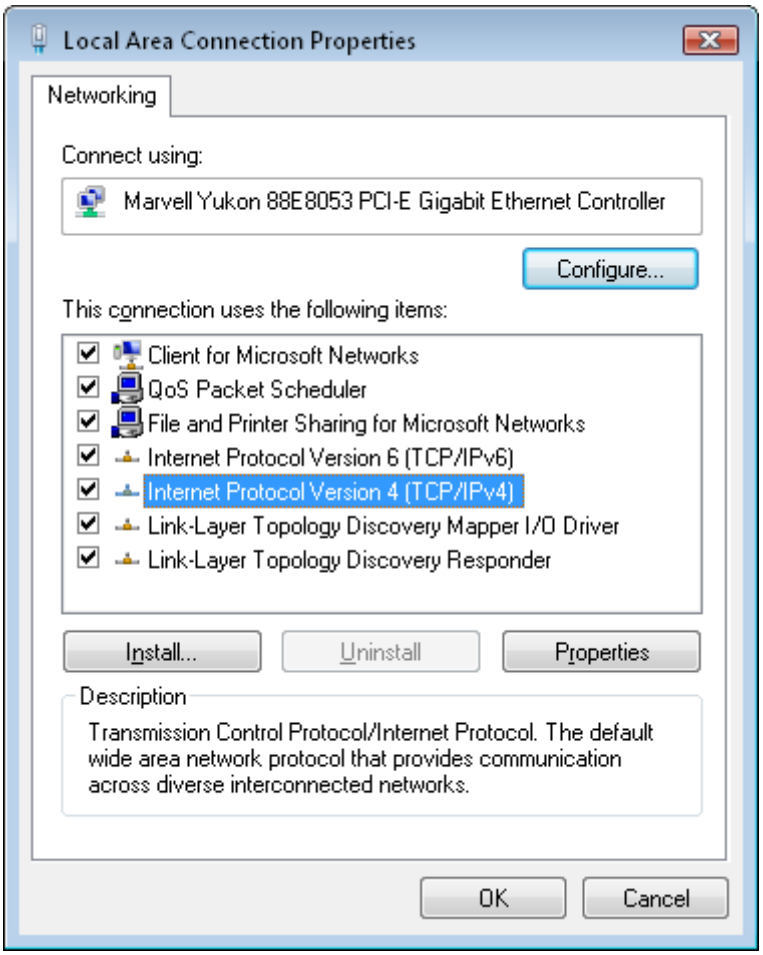
There are several ways to set up multiple IP addresses on a computer:

1. To have multiple network interface cards (NICs) on your computer and to assign a different IP address to each card.
2. To assign multiple IP addresses to a single NIC.
3. To combine 2 previous options: have multiple NICs with multiple IPs assigned to one or more of them.

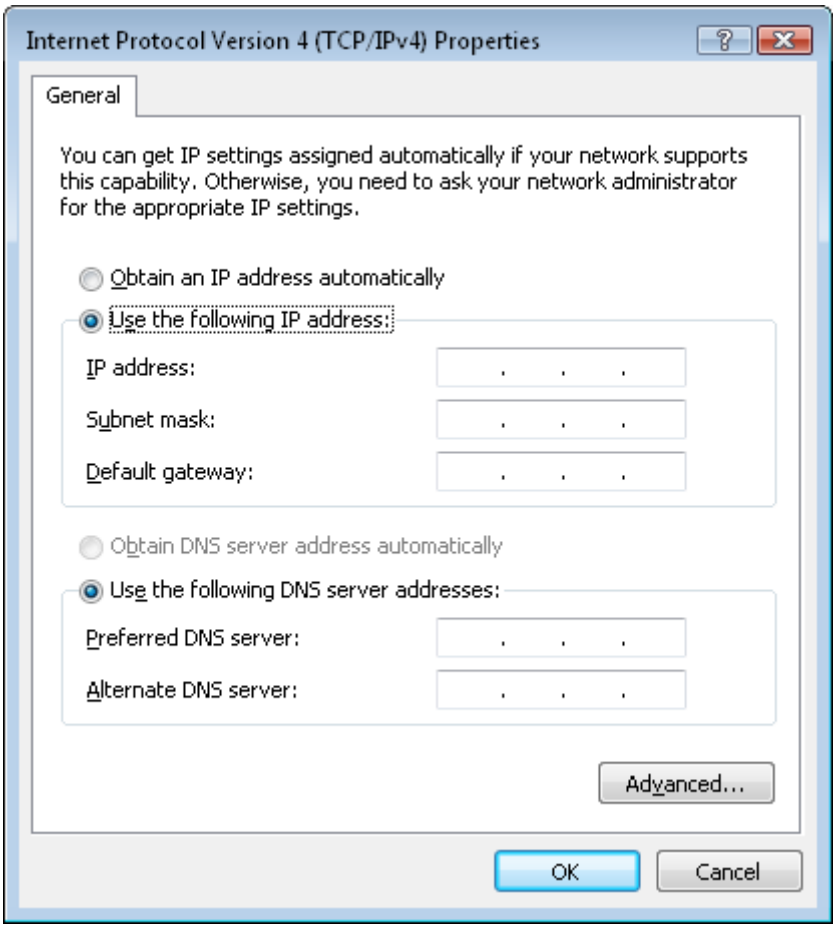
By default, each network interface card (NIC) has its own unique IP address. However, you can assign multiple IP addresses to a single NIC.

If you want to assign more than one IP address to a network card on Windows Vista, follow the steps below.

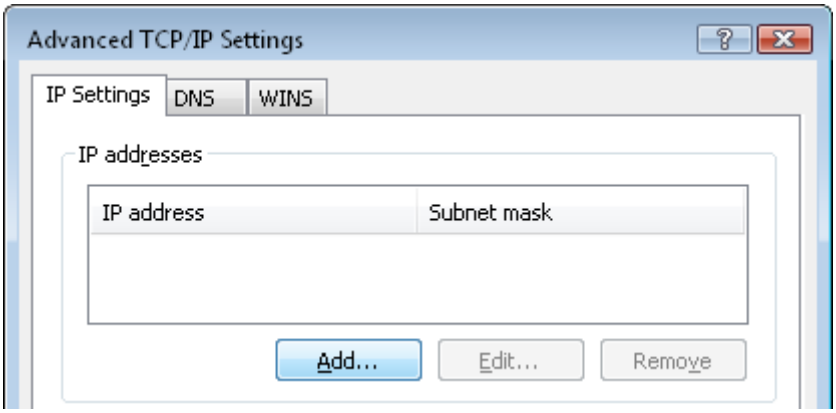
1. Choose **Settings -> Network Connections** on the Windows **Start** menu.
2. Right-click on the **Local Area Connection**, choose **Properties**.



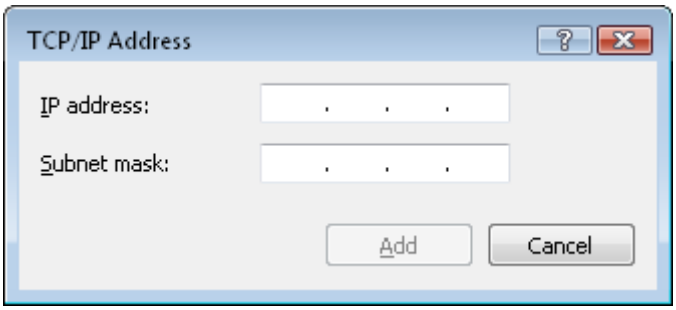
3. Highlight **Internet Protocol (TCP/IP)**, click **Properties**.



4. If you use **DHCP**, you should disable it: click **Use the following IP address** and enter IP address, Subnet mask and Default gateway.
5. Click **Advanced...** at the bottom.



6. Enter additional IP addresses: click the **Add...** button and enter a new IP address and Subnet mask.



If you use Windows XP, the whole procedure will be the same, except for the first steps:

- Right-click on **My Network Places**, choose **Properties**.
- Right-click on the **Local Area Connection**, choose **Properties**.